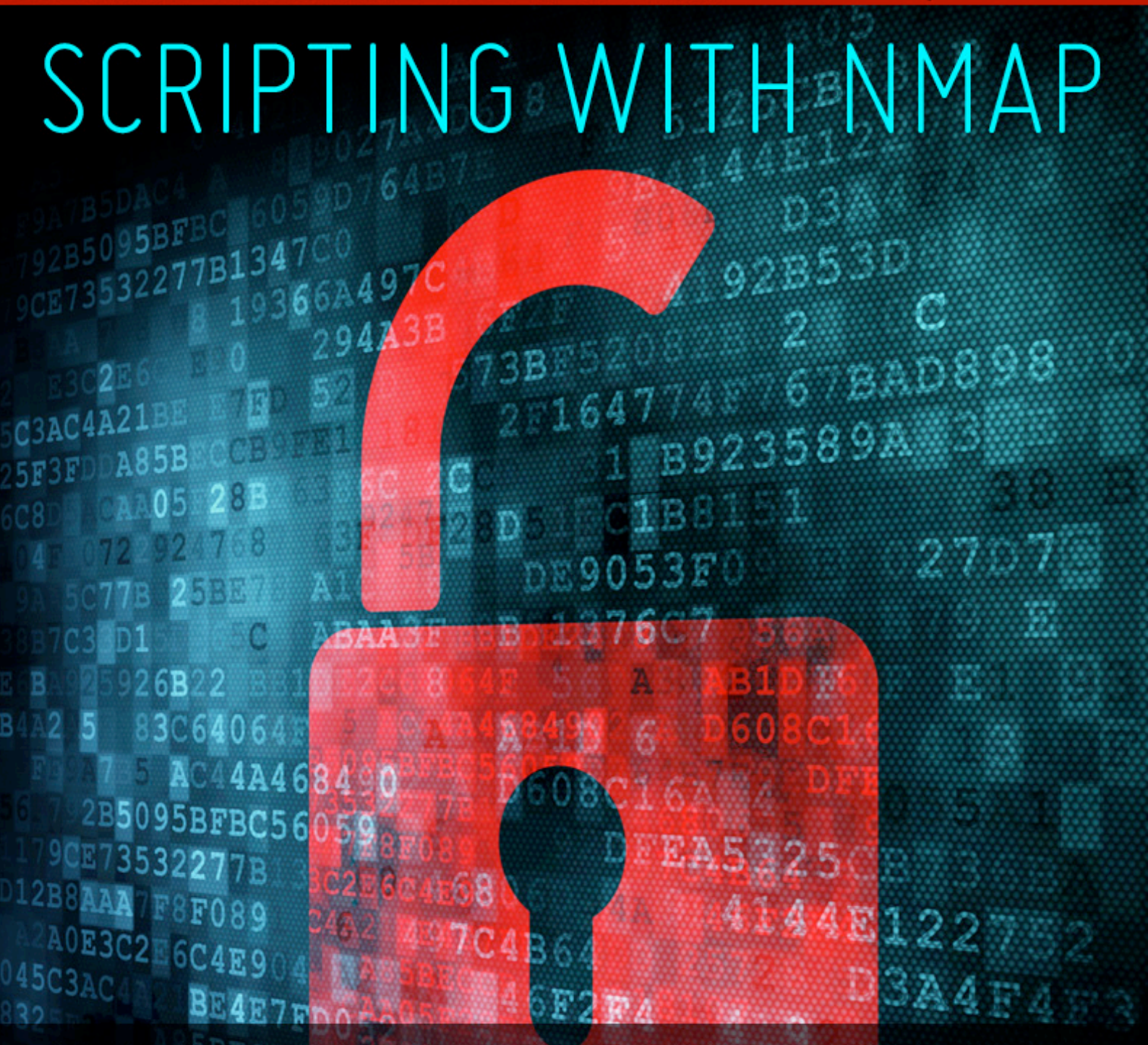
[IN]SECURE

OPEN. INFORMATIVE. TO THE POINT. Issue 35 - September 2012

SCRIPTING WITH NMAP



EVIL APPLICATIONS OF AUGMENTED REALITY

MONITOR THE BLIND SPOTS IN YOUR IT SYSTEM

HURDLES TO SECURITY AND COMPLIANCE IN INDUSTRIAL CONTROL SYSTEMS

RSA CONFERENCE EUROPE 2012

9-11 OCTOBER | HILTON LONDON METROPOLE | U.K.

PRACTICAL SOLUTIONS TO HEADLINE THREATS.

Three days of information security insight.

Only RSA® Conference Europe 2012 delivers the steps and strategies needed to protect your organisation's assets. From managing smartphones and tablets, to the workplace risks from social media tools, get the techniques you want and the answers you need.

Hear from highly regarded keynotes including Wikipedia founder Jimmy Wales, internationally renowned security technologist Bruce Schneier, and investigative journalist, author and broadcaster Misha Glenny – one of the world's leading experts on cybercrime and global mafia networks.

- Leave with actionable solutions
- Build your skills
- Network with like-minded professionals
- Stay informed, stay ahead

Get the practical insight your organisation needs. Attend and play your part in Europe's most informative information security event.

Full Session Agenda now online.

Date: 9 - 11 October
Venue: Hilton London Metropole Hotel, U.K.

Hear how the world's security experts manage challenges like:

- Mobile security
- Data breaches
- Hacktivism
- Cybercrime
- Malware threats
- Cloud computing

THE GREAT CIPHER
MIGHTIER THAN THE SWORD

Find out more at

www.rsaconference.com/help

TABLE OF CONTENTS

Depending on where in the world you are, nice weather and holidays are behind us as we come back to our offices. Now is the perfect time to enjoy another issue of (IN)SECURE, filled with a variety of topics for all knowledge levels. We have Didier Stevens illustrating administrative Nmap scripting, Greg Conti and colleagues discussing evil applications of augmented reality, Gavin Watson talking about social engineering attacks, and much more.

Next month we'll be in London for RSA Conference Europe 2012 and I'm looking forward to stimulating conversations and provocative talks. If you're attending, get in touch!

Mirko Zorz
Editor in Chief

Visit the magazine website at www.insecuremag.com

(IN)SECURE Magazine contacts

Feedback and contributions: Mirko Zorz, Editor in Chief - mzorz@net-security.org
News: Zeljka Zorz, Managing Editor - zzorz@net-security.org
Marketing: Berislav Kucan, Director of Operations - bkucan@net-security.org

Distribution

Google adds Do Not Track support to Chrome

The next official release of Google's Chrome browser will more than likely support the Do Not Track (DNT) initiative by sending the DNT HTTP header to websites if the user chooses it, as the support was added to the browser's latest developer build.

With this move, Chrome becomes the last of the "big" browsers to implement it - until now Chrome users who wanted to have the option were required to download and use an official Do Not Track add-on.

The Do Not Track initiative is endorsed by the FTC, and the project includes collaborators from technology companies, privacy advocacy groups, and a number of independent researchers. The Do Not Track header enables users to opt out of tracking by websites they do not visit, and that includes social platforms, advertising networks, and analytics services.

Since Google has a major stake in the market of online advertising, it is understandable that the company delayed incorporating the option in its own browser.

At the time being, websites are not required to comply with the user's Do Not Track request, so it offers little protection. Of the large Internet companies out there, only Twitter has voiced its support for the initiative and has rolled out the DNT opt-out cookie.

In the meantime, Microsoft has announced that the new Internet Explorer 10 will have "Do Not Track" on by default, and Roy Fielding - one of the founders of the Apache HTTP Server Project, a scientist at Adobe and one of the editors of the DNT standard - reacted to the news by adding a patch to the open source Apache HTTP Server that will make it ignore the DNT header if sent by the IE10 browser.

Best practices for mobile software developers

The PCI Security Standards Council (PCI SSC), a global, open industry standards body providing management of the Payment Card Industry Data Security Standard (PCI DSS), PINTransaction Security (PTS) requirements and the Payment Application Data Security Standard (PA-DSS), released best practices for mobile payment acceptance security.

The PCI Mobile Payment Acceptance Security Guidelines offer software developers and mobile device manufacturers guidance on designing appropriate security controls to provide solutions for merchants to accept mobile payments securely.

The document organizes the mobile payment-acceptance security guidance into two categories: best practices to secure the payment transaction itself, which addresses cardholder data as it is entered, stored and processed using mobile devices; and guidelines for securing the supporting environment, which addresses security measures essential to the integrity of the broader mobile application platform environment.

Key recommendations include:

• Isolate sensitive functions and data in trusted environments
• Implement secure coding best practices
• Eliminate unnecessary third-party access and privilege escalation
• Create the ability to remotely disable payment applications
• Create server-side controls and report unauthorized access.

New ISO 27001 & ISO 22301 Documentation Toolkit launched

Information Security & Business Continuity launched a new version of its ISO 27001 & ISO 22301 Documentation Toolkit. This new version is available in five languages and is fully compliant with the new ISO 22301 standard.

The new version has some significant improvements: it has been aligned with new international business continuity standard ISO 22301, but also it now includes several video tutorials and webinars on demand that help fill in the documentation and implement it in day-to-day operations.

"With this Toolkit we wanted to ease the pain of ISO 27001 and ISO 22301 implementation - according to the feedback of our clients worldwide, they have saved up to 50% of time and 30% of implementation costs," says Dejan Kosutic, the author of the Toolkit. He also added, "This is because we took special care not to create overhead for our clients - we developed an optimum number of documentation templates, and provided hands-on advice on how to implement the documentation."

The Toolkit contains 59 documentation templates in MS Word and Excel, and access to 12 video tutorials and 16 webinars on demand. In each template it is clearly marked where the company needs to input specific information like company name, responsibilities, etc., and comments throughout the templates explain available options for the document text.

Detailed specifications for this product can be found at www.iso27001standard.com/en/services/iso-27001-bs-25999-premium-documentation-toolkit

Microsoft issues workaround for IE 0-day exploited in current attacks

Microsoft has issued a security advisory with advice on how to patch a Internet Explorer zero-day vulnerability recently spotted being exploited in the wild by attackers that might be the same ones that are behind the Nitro attacks.

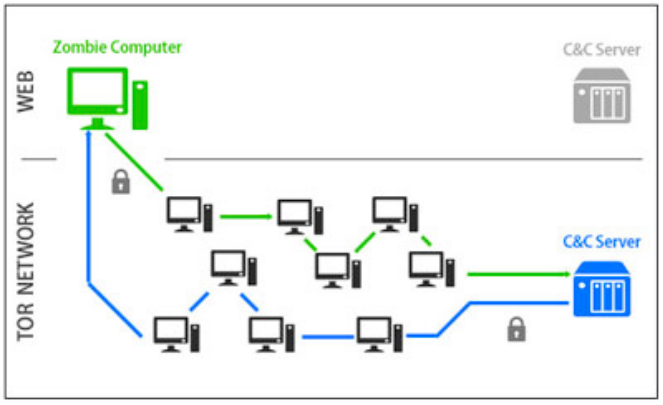
The existence of the flaw and a working exploit for it has been revealed by security researcher and Metasploit contributor Eric Romang, who discovered it on 14 September while monitoring some of the infected servers used by the Nitro gang in the recent Java attacks.

The Rapid7 team got right on it and created a module exploiting the vulnerability for the Metasploit exploit toolkit during the weekend, and advised IE users to switch to other browsers such as Chrome or Firefox until Microsoft patches the flaw security update becomes available.

Microsoft has reacted fast by issuing a security advisory in which it confirms the existence of the flaw in Internet explorer 9 and all previous versions (IE10 is not affected), and offers instructions on steps the users can take to mitigate - but not yet remove - the threat.

These steps could bring additional problems to the users, such as being bombarded by a slew of security warnings, so until Microsoft releases a definitive patch for the hole, maybe it would be easier for IE users to take Rapid7's advice and switch to another browser for the time being.

Botnet operators hide C&Cs in the Tor network



Over the years, botnet owners have tried out different tactics for keeping their C&C servers online, in contact with the zombie computers, and hidden from researchers and law enforcement agencies.

The location of a centralized C&C server could be concealed by everyday domain-changing, but the algorithm that does that can be reverse engineered. Once the location is established, the server's takedown leaves the bots orphaned. A Peer-to-Peer architecture can solve the aforementioned problem of the single point of failure by making every zombie a kind of C&C server and capable of issuing commands to others. Still, the problems with this approach are many: routers blocking incoming traffic, protocols that must be especially designed for respective bots, and the possibility of an easy takeover of the botnet by law enforcement agencies or other bot herders.

A third, more fitting solution has been discovered by GData Software researchers, who spotted a botnet with its C&C server hidden behind the layers of the Tor anonymity network.

The advantages are many - the server is anonymous and can't point to the botnet owners' identity, and by the same token, can't be taken down easily. The traffic to and from the server is encrypted by Tor, so IDS solutions can't block it. In fact, blocking Tor traffic in general is not usually done, because there are a lot of legitimate uses for it. Finally, the bot creator does not have to create a custom protocol but, as it is in this particular case, can use the existing and reliable IRC protocol. Unreliability and sluggishness are what makes this approach less than ideal, but the pros definitely outweigh the cons.

Chip and PIN payment card system vulnerable to "pre-play" attacks

A team of Cambridge University researchers has recently discovered that a flaw in the way that the algorithms for generating unique numbers for each ATM or POS transaction are implemented makes it possible for attackers to authorize illegal transactions without ever having to clone the customers' card.

"The UN (unique number) appears to consist of a 17 bit fixed value and the low 15 bits are simply a counter that is incremented every few milliseconds, cycling every three minutes," they discovered.

"We wondered whether, if the 'unpredictable number' generated by an ATM is in fact predictable, this might create the opportunity for an attack in which a criminal with temporary access to a card (say, in a Mafia-owned shop) can compute the authorization codes needed to draw cash from that ATM at some time in the future for which the value of the UN can be predicted."

Their research led them to conclude that the number in question is predictable, and that such a "pre-play" attack - while not that easy to execute and possessing certain limitations - is possible and viable in practice through a number of approaches, which include malware-infected ATMs, supply chain attacks, terminal cut-out, UN modification in the network, and the cooperation of a merchant.

Selected banks, payment switches and major card companies have been informed of the vulnerability, but most refused to formally comment on the findings.

39% of IT staff can get unauthorized access to sensitive information

IT professionals are allowed to roam around corporate networks unchecked, according to a survey of more than 450 IT professionals by Lieberman Software.

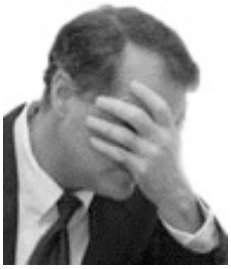
It found that 39% of IT staff can get unauthorized access to their organization's most sensitive information – including the CEO's private documents – and one in five has already accessed data they shouldn't.

The study found that, if they thought their job was at risk, 11% of respondents would abuse their administrative rights to snoop around the network to seek out the redundancy list and other sensitive information. In fact, if laid off tomorrow, 11% would be in a position to take sensitive information with them. Worryingly, nearly a third confirmed that their management does not know how to stop them.

Organizations can control privileged account access, and diminish the insider threat, with a four-part process:

- Identify and document critical IT assets, their privileged accounts and their interdependencies.
- Delegate access to privileged credentials so that only appropriate personnel, using the least - privilege required, can login to IT assets.
- Enforce rules for password complexity, diversity and change frequency, and synchronize changes across all dependencies.
- Audit and alert so that the requester, purpose, and duration of each privileged access request is documented.

Publishing firm says leaked Apple IDs came from their servers



BlueToad, a Florida-based digital edition publishing company, has announced that the recent massive Apple UDID leak originated from their own servers, and not an FBI laptop.

They were first alerted to the possibility by David Schuetz, a researcher employed by mobile device security consulting firm Intrepidus Group, who took the trouble to analyze the leaked UDIDs and the device names attached to them.

After discovering that a considerable number of devices had names that referenced Blue Toad and seemed to belong to the company's various departments, and after noticing that these devices' UDIDs showed up again and again in the list, he was pretty sure that they could be involved somehow.

"As soon as we found out we were involved and victimized, we approached the appropriate law enforcement officials, and we began to take steps to come forward, clear the record and take responsibility for this," BlueToad's CEO Paul DeHart shared with NBC.

The investigation discovered that the data had been stolen from the company's servers at some point during the last few weeks, but no details about how it happened have been released.

DeHart pointed out that they, of course, can't be sure that once the information was stolen from their servers hasn't ended up on a FBI laptop, but says that one thing they definitely do know is that it didn't contain 12 million UDIDs.

BEAST developers come up with new SSL/TLS attack

From the security researchers who created and demonstrated the BEAST (Browser Exploit Against SSL/TLS) tool for breaking SSL/TLS encryption comes another attack that exploits a flaw in a feature in all versions of TLS.

Dubbed CRIME by the researchers Juliano Rizzo and Thai Duong, the attack works similarly to the BEAST attack, and will be presented for the first time to the public during the ekoparty Security conference which is to be held in Argentina.

Without wanting to reveal much about the soon-to be unveiled attack, the researchers shared that the feature in question leaks information that can be used to decrypt user cookies, extract the login information contained in them and hijack their sessions.

"By running JavaScript code in the browser of the victim and sniffing HTTPS traffic, we can decrypt session cookies. We don't need to use any browser plug-in and we use JavaScript to make the attack faster but in theory we could do it with static HTML," Rizzo explained to ThreatPost.

This new attack is more effective than the BEAST, as the latter affected only TLS 1.0 and SSL 3.0 and could be foiled by simply switching to other versions of the standard and from using the AES algorithm to employing the RC4 one.

The CRIME, unfortunately, affects all SSL/TLS versions and both Firefox and Chrome are vulnerable to the attack, although the researchers have notified Mozilla and Google about the flaw and patches are expected to be released soon.

Goodbye Olympics, hello SANS London

As summer ends, and winter isn't such a long way off; now is the ideal time to start thinking about Europe's biggest information security training event.

Running from 26 November to 3 December, SANS London 2012 will offer 13 of SANS' top, cutting-edge, hands-on InfoSec training courses taught by the best instructors in the industry.

On top of this fantastic lineup of classes, we will also be bringing the extras that SANS London is famous for: NetWars, evening talks, and other exciting learning and networking opportunities.

If you register and pay before 10 October, you will receive a 400 Euro discount to your tuition fees, see www.sans.org/info/109869.

Zero-day-loving Google hackers furiously active in last three years

Symantec has released a research paper that details a three years' worth of attacks that can all be tracked back to a single large group - the very group that was behind the Aurora attacks - that continuously uses components of an infrastructure the researchers have dubbed the "Elderwood platform" after a parameter used in the attack codes.

The Elderwood gang is primarily interested in gathering and stealing intelligence (trade secrets, contacts, infrastructure details, intelligence for future attacks) and intellectual property (designs and plans) from an ever-increasing number of companies mostly located in the United States.

These companies are usually defense supply chain manufacturers, human rights and non-governmental organizations, and IT service providers. But the thing that makes the Elderwood gang really stand out from other players in the cyber espionage field is that they seemingly have an inexhaustible supply of zero-day exploits at their disposal - they have used eight in the last three years.

"In order to discover these vulnerabilities, a large undertaking would be required by the attackers to thoroughly reverse-engineer the compiled applications. This effort would be substantially reduced if they had access to source code," the researchers pointed out.

This last theory sounds be the most likely, as the gang has hit a number of software companies in the last year. As mentioned before, Adobe has been hit around the same time as Google, and it seems probable that the attackers are the same ones, i.e. the Elderwood gang.

It's possible that they got their hands on source code for Adobe's products, and have managed to reverse-engineer them and have, therefore, an easier time finding out zero-day vulnerabilities in them. All the exploits used so far targeted two of the most popular applications out there: Microsoft Internet Explorer and Adobe Flash Player.

Once the exploit compromises the targeted computer, a backdoor or a dropper Trojan (including the Hydraq/Aurora Trojan that has been used in the attack against Google) is delivered and set up on it, allowing the attackers stealthy and continuous access, or the ability to download other malware on the machine.

The Elderwood gang uses two primary attack vectors: spear phishing emails sent to specific targets, and so-called watering hole attacks - the compromise of websites targets are likely to visit and equipping them with iframes pointing to a server hosting exploits for the zero-days.

The sheer number of attacks, the skill-set wielded by the attackers and the choice of targets all seem to point to a nation state, or a group backed by a nation state, although it is also possible that a large and well-founded criminal gang might be behind the attacks.

Nmap deserves a place in your administrator's toolbox. Do you believe Nmap is only useful to network engineers and hackers? Think again. This article will show you how Nmap scripts can ease your life as a system administrator.

The Nmap (Network Mapper) open source network scanner was first released in 1997. Since then, each major Nmap release came with impressive new features.

An important milestone for us was the addition of the Nmap Scripting Engine (NSE) to Nmap in 2006. 2012 saw a new major release: version 6.

Users can extend Nmap by writing scripts for the Nmap Scripting Engine, for which the Nmap developers choose the Lua programming language. Lua (from the Portuguese word for “moon”) is an imperative and object-oriented programming language.

If you know how to program with an imperative programming language, programming with Lua will feel very familiar. Lua has functions and flow control statements like if, while and for. If you are new to programming, Lua is a good language with which to start because of its gentle learning curve.

The administrative problem we want to address and solve with the Nmap Scripting Engine is detecting and identifying the McAfee ePolicy Orchestrator (ePO) agent. The McAfee ePolicy Orchestrator is a security management platform. One of the products it manages is McAfee VirusScan Enterprise installed on clients.

ePO servers manage clients via the ePO agent, software that is installed on each client that has to be managed by the ePO server. By default, this ePO agent listens on TCP port 8081. A simple Nmap scan on port 8081 allows us to identify machines without ePO agent, and thus not under control of the ePolicy Orchestrator.

These machines do not have an open TCP port 8081. So there is no need to write a script to find machines without ePO agent, we can do this with standard Nmap. But identifying machines with an ePO agent is trickier.

To successfully identify a machine with ePO agent, we need to find TCP port 8081 open, and it has to reply to an appropriate HTTP GET request with an XML reply that contains the ePO agent's properties. Performing an HTTP GET request and interpreting the XML reply requires scripting.

When we navigate with a web browser like Microsoft Internet Explorer to http://10.0.0.2:8081, we will get the log file of the ePO agent running on machine testserver. This is actually an XML document transformed with an XSL script:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="FrameworkLog.xsl"?>
<naLog>
  <ComputerName>TESTSERVER</ComputerName>
  <version>4.5.0.1852</version>
  <AgentGUID>{D2E157F4-B917-4D31-BEF0-32074BADF081}</AgentGUID>
  <ePOServerName>EPOSERVER</ePOServerName>
  <Log component="8192" time="2012-06-14T11:01:48" type="3">Agent is looking for events to upload</Log>
  <Log component="8192" time="2012-06-14T11:03:47" type="3">Agent is looking for events to upload</Log>
  <Log component="8192" time="2012-06-14T11:05:47" type="3">Agent is looking for events to upload</Log>
  <Log component="8192" time="2012-06-14T11:07:47" type="3">Agent is looking for events to upload</Log>
  <Log component="8192" time="2012-06-14T11:09:11" type="3">Agent Started Enforcing policies</Log>
  <Log component="2" time="2012-06-14T11:09:11" type="3">Enforcing Policies for McAfee Agent</Log>
```

The following NSE script is the minimum we need to perform an HTTP GET request to TCP port 8081:

```
require "http"
require "shortport"

description = [[
Check if EPO agent is running on port 8081
]]
author = "Didier Stevens (https://DidierStevens.com)"
license = "Source code put in public domain by Didier Stevens, no Copyright"
categories = {"discovery", "safe"}
portrule = shortport.portnumber(8081, "tcp")
action = function(host, port)
    http.get(host, port, '/')
end
```

In this script, the first two require statements load libraries http and shortport we need later on in our script. Then we have four fields: description, author, license and categories.

Field description defines a string that contains the description of the string. Double brackets ([[]]) are used in Lua to enclose strings that span multiple lines.

Fields author and license are strings that identify the author and the license type respectively. Double quotes are used in Lua to enclose a string on a single line.

Field categories is a table of strings that define to which categories the script belongs to. A discovery script tries to find out more information about the target, and a safe script will not have a negative impact on the machine it is targeting.

The portrule function is a rule that determines whether the script should be run against a target or not. With shortport.portnumber(8081, "tcp"), we define that the script should run when TCP port 8081 is open.

And finally, we have the heart of the script, the action function. In our script, this function performs a HTTP GET request for path / against our target (defined with the arguments host and port).

To execute this script, we save it with the name mcafeeepoagent.nse into Nmap's folder scripts and launch Nmap like this:

```
nmap -p8081 --script
mcafeeepoagent.nse 10.0.0.2
```

This is the core of our script. But if we examine the result we get for our HTTP GET query, we see it is not XML. That is because the ePO agent will only reply with XML if the query is done by a browser like Microsoft Internet Explorer. To make our script spoof Internet Explorer, we need to change the User-Agent string like this:

```
options = {header={}}
options['header']['User-Agent'] = "Mozilla/4.0 (compatible; MSIE 6.0; Windows
NT 5.0; mcafeeepoagent)"
data = http.get(host, port, '/', options)
```

We define an options table and set the User-Agent string, then pass options as an argument to the http.get function. The reply to our request is in the data variable.

Next we need to check if we actually got a reply with a body, and if the body contains the type of XML an ePO agent would produce. We do this we the following if statements:

```
if data.body then
    if data.body:StartsWith('<?xml version="1.0" encoding="UTF-8"?><?xml-
-stylesheet type="text/xsl" href="FrameworkLog.xsl"?><naLog>') then
```

StartsWith is a method we added to the string class like this:

```
function string.StartsWith(stringToSearch, stringToFind)
    return stringToFind == stringToSearch:sub(1, #stringToFind)
end
```

Method StartsWith does what its name indicates: it tests if a given string starts with a specified string.

Next, we need to extract relevant information from the XML document. This information is found in XML elements ComputerName, version, AgentGUID and ePOServerName. The following function uses regular expression matching to find the content of an XML element:

```
function ExtractXMLElement(xmlContent, elementName)
    return xmlContent:match("<" .. elementName .. ">([^<]*)</" .. elementName
.. ">")
end
```

Function ExtractXMLElement takes the XML document and the name of the XML element as arguments and returns the content of the XML element. This allows us to extract the information and return it like this:

```
computerName =  ExtractXMLElement(data.body, "ComputerName") or ""
epoServerName = ExtractXMLElement(data.body, "ePOServerName") or ""
version =       ExtractXMLElement(data.body, "version") or ""
agentGUID =     ExtractXMLElement(data.body, "AgentGUID") or ""

return string.format("ePO agent found,%s,%s,%s,%s", computerName, version,
epoServerName, agentGUID)
```

If one of the XML elements is missing, we use the empty string “”.

And if we do not find the XML data we expect, we return this fact like this:

```
return "ePO agent not found"
```

Here is the complete script:

```
-- mcafeeepoagent.nse V0.0.1, checks if ePO agent is running
-- Source code put in public domain by Didier Stevens, no Copyright
-- https://DidierStevens.com
-- Use at your own risk
--
-- History:
--   2012/05/31: Start
--   2012/06/01: extracting data from XML; tested with ePO 4.5 and 4.6

-- http://nmap.org/svn/docs/sample-script.nse

description = [[
Check if ePO agent is running on port 8081
]]

---
--@output
-- Nmap scan report for testserver (192.168.1.1)
-- Host is up (0.00s latency).
-- rDNS record for 192.168.1.1: testserver.local
-- PORT      STATE SERVICE
-- 8081/tcp open  unknown
-- |_mcafeeepoagent: ePO agent found:
TESTSERVER,4.5.0.1852,EPOSERVER,D2E157F4-B917-4D31-BEF0-32074BADF081
-- MAC Address: 00:11:22:33:44:55 (Apple Computer)

author = "Didier Stevens (https://DidierStevens.com)"

license = "Source code put in public domain by Didier Stevens, no Copyright"

categories = {"discovery", "safe"}

require "http"
require "shortport"

portrule = shortport.portnumber(18081, "tcp")

function string.StartsWith(stringToSearch, stringToFind)
    return stringToFind == stringToSearch:sub(1, #stringToFind)
end

function ExtractXMLElement(xmlContent, elementName)
    return xmlContent:match("<" .. elementName .. ">([^<]*)</" .. elementName
.. ">")
end

action = function(host, port)
    local options, data, computerName, epoServerName, version, agentGUID

    -- Change User-Agent string to MSIE so that the ePO agent will reply with
XML
```

```
    options = {header={}}
    options['header']['User-Agent'] = "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.0; mcafeeepoagent)"
    data = http.get(host, port, '/', options)

    if data.body then
        if data.body:StartsWith('<?xml version="1.0" encoding="UTF-
-8"?><?xml-stylesheet type="text/xsl" href="FrameworkLog.xsl"?><naLog>') then
            computerName =  ExtractXMLElement(data.body, "ComputerName") or
""
            epoServerName = ExtractXMLElement(data.body, "ePOServerName") or
""
            version =       ExtractXMLElement(data.body, "version") or ""
            agentGUID =     ExtractXMLElement(data.body, "AgentGUID") or ""

            return string.format("ePO agent found,%s,%s,%s,%s", computer-
Name, version, epoServerName, agentGUID)
        end
    end

    return "ePO agent not found"
end
```

When we use this script against a machine with an ePO agent, Nmap produces output like this:

```
Nmap scan report for testserver (10.0.0.2)
Host is up (0.00s latency).
rDNS record for 10.0.0.2: testserver.local
PORT      STATE SERVICE
8081/tcp open  unknown
|_mcafeeepoagent: ePO agent found:
TESTSERVER,4.5.0.1852,EPOSERVER,D2E157F4-B917-4D31-BEF0-32074BADF081
MAC Address: 00:11:22:33:44:55 (Apple Computer)
```

And if the TCP port is opened but by a service other than an ePO agent, the output is like this:

```
Nmap scan report for testserver (10.0.0.2)
Host is up (0.00s latency).
rDNS record for 10.0.0.2: testserver.local
PORT      STATE SERVICE
8081/tcp open  unknown
|_mcafeeepoagent: ePO agent not found
MAC Address: 00:11:22:33:44:55 (Apple Computer)
```

I hope this article will inspire you to start writing your own Nmap Scripting Engine scripts for your administrative tasks. You can take the first example as a template to get you started.

And do not forget that in order to execute your script, you not only need to call your script, but the targeted port needs to be open.

Didier Stevens (Microsoft MVP Consumer Security, CISSP, GSSP-C, MCSD .NET, MCSE/Security, MCITP Windows Server 2008, RHCT, CCNP Security, OSWP) is an IT Security Consultant currently working at a large Belgian financial corporation. He is employed by Contraste Europe NV, an IT Consulting Services company (www.contraste.com). You can find his open source security tools on his IT security related blog at blog.DidierStevens.com.

What has been your biggest challenge in the role of Executive Director of ENISA? How has your background helped shape your role in the organization?

Probably the greatest challenge for me at ENISA is to increase the Agency's visibility and to create a bigger impact with our work. The entire ENISA team puts maximum effort into identifying how we can improve cyber security in Europe and making sure that we communicate our knowledge as effectively as possible. We're constantly looking at how we get our message across to everyone who has a stake in network and information security. There's a huge potential audience for us to reach, and it's one that is constantly growing as information technology (IT) touches the lives of an ever-increasing number of Europe's citizens.

ENISA is a relatively small organization - about 60 people, including directly employed staff, experts seconded from EU Member States, contractors and agency personnel - and while our work supports IT security all across Europe, we focus on issues where our expertise can make a real difference, whether that means giving advice to Member States on good practice in safeguarding critical information infrastructure, or in advising on new European laws to ensure that sensitive information stays protected.

A large part of ENISA's work relates to bringing together the various organizations and individuals involved in network and information security. Information technology plays a role in practically every aspect of our lives, and staying abreast of this constantly evolving picture is another big, but very enjoyable challenge.

I have worked in a number of different spheres over the past 25 years, and this has allowed me to experience and be part of how different organizations operate and shape themselves to meet their varying tasks. My experience includes working in the aerospace industry, with DASA/MBB, starting out in systems analysis, and ending as the company's Programme Manager for IT.

I have also worked in the insurance industry, and in the academic world. Immediately before joining ENISA in 2005, I was President of Germany's Federal Office for information

Security (BSI) for six years. While there, I took the lead in building a cooperation between BSI and the IT security industry, and was also responsible for raising public awareness about information security issues.

I would say that my background has helped me to develop an approach focused on analyzing the organization's objectives, and constantly seeking to find the best structure to meet them. The way the Agency (or any other organization) operates needs to be in tune with the challenges the Agency faces. For example, we often need to be "on the spot" at a short notice to provide support to Member States.

To support this, last September we launched a Mobile Assistance Team (MAT) that operates out of our Athens branch office. This small unit consisting of four people has clocked up more than 40 assistance missions within the year to support the Member States in addressing security issues. This flexible, responsive approach is also favored by Neelie Kroes (the European Commissioner for Digital Agenda) and we plan to do more work in this way.

THE PAST YEAR HAS SEEN GREAT SUCCESS IN INTERNATIONAL COOPERATION ON CYBER SECURITY EXERCISES

When taking into account all the research that ENISA has done in the past year, how would you rate the current state of information security in Europe? What key areas still need work?

There has been a lot of progress in IT security over the past year. ENISA's work on this has included supporting new Computer Emergency Response Teams (CERTs) that have been set up in Romania, Malta and Ireland, and we've done a great deal of work to help CERTs in all Member States build stronger links and share good practices. For example, there is our annual CERT workshop, which last year included Europol as a participant, so that we could bring the dimension of cybercrime into the picture. This type of knowledge sharing is crucial.

The past year has also seen great success in international cooperation on cyber security exercises. In 2011, ENISA worked with the European Commission, Member States and the US to hold the first ever EU-US cyber security exercise, Cyber Atlantic 2011. This was built on the experience we gained from facilitating the first ever pan-European exercise in 2010, and we are currently finalizing arrangements for Cyber Europe 2012, which is scheduled for this autumn. (The exact date is kept confidential for security reasons.)

More widely, with Article 13a of the Commission's Telecomms Regulation, we have seen moves towards standardized reporting of data security breaches, and steps are being made towards a common IT security governance structure for the EU.

Of course, there is still much work to be done on consolidating knowledge and building shared approaches and understanding in all of these areas. In addition, there are newer, emerging areas that offer great opportunities for us as users of information technology, but also require an understanding of new security challenges.

Cloud computing, for example, offers great savings in terms of cost end efficiency, but needs to be implemented with due regard to data security requirements. The laws and legal regimes of countries that are hosting data need to be known, and care must be taken to ensure that adequate legal safeguards are in place.

Another emerging area is smart grids, which can provide more efficient use of power networks. However, the relationships between new, Internet-based technologies and existing traditional control systems that are now becoming "embedded" in the Internet need to be understood to ensure that they do not create a vulnerable point that could be used as a way in for cyber attackers. Recent ENISA reports have looked at both cloud computing and smart grids, and these will be areas of ongoing work for us.

Which European countries are excelling when it comes to computer security? What actions are they taking and how can those be an example for other members of the European Union?

There are so many factors involved in network and information security that it's difficult to draw comparisons between countries. In any event, different Member States have had different evolutions in the way they have developed their IT infrastructure and policies, so their security requirements can be very different.

Having said that, countries that have long-established IT and telecoms infrastructures and home-based IT industries also tend to have well-developed and mature strategies for security. One of ENISA's core activities is to facilitate the sharing of good practices, and we actively work to find Member States that have expertise in a particular area that they are willing to share with countries that are keen to learn from the experience of others.

While some consider compliance to be an essential step towards greater security, others largely dismiss it as an expensive step that yields a false sense of security. What is your take on compliance and its influence on information security in Europe?

Compliance is essential. Laws and regulations are developed and standards are put in place so that consumers and businesses can be assured that protection is in place, and that legal remedies are available if anything goes wrong.

Of course, that does not mean that players in the IT or telecoms field should do nothing more than comply with legal minimums, or wait for legislation to push them towards implementing good security practices. Providers can themselves do much to anticipate and take measures against cyber attacks. This can include actions they take by themselves and recommendations for customers on how to stay secure online. Again, ENISA has developed and offers guidance on these areas.

In parallel with compliance, ENISA is working with the Commission to further develop public-private partnerships (PPPs) under the European Public Private Partnership for Resilience (EP3R) programme.

This works by ENISA establishing trusted information sharing relationships with national PPPs and then disseminating that knowledge more widely. We've produced a good practice guide on this, and can, on request, also assist in developing a national PPP by, for example, providing strategic and technical advice at the planning, establishment and execution phases.

When taking into account all that can happen, a nation's critical infrastructure is fragile and in serious need of protection. In an era of cyber attacks, concerns grow even more. What should be done in order to make Europe's smart grid attack proof?

As I mentioned briefly earlier with regard to smart grids, they can give rise to new information security challenges for electricity networks. Vulnerabilities can be exploited to disrupt networks or even shut down power plants for financial or political motivation. This is reported to have happened in 2009, when US officials recognized that cyber spies had hacked into the US electricity grid. This makes both the software and hardware for smart grid infrastructure high-value and high-risk targets.

In a report earlier this year, ENISA looked at smart grids, and concluded that the two "separate worlds" of the energy and IT security sectors must be aligned to achieve security. We estimate that without taking cyber security into serious consideration, smart grids may evolve in an uncoordinated manner.

I would therefore suggest that smart grids' security be made part of the EU's forthcoming Internet Security Strategy, and we recommend that The European Commission and Member States provide a clear regulatory and policy framework at EU and national level – something that is currently missing. We also suggest that ENISA collaborate with Member States and the private sector on developing a minimum set of safety guidelines based on existing standards. Other steps should include the promotion of a security certification scheme for the entire value chain of smart grid components and organizational security.

Member States should also take advantage of existing capabilities. Smart grids are a relatively new development, so there is the opportunity to build security into them from the outset.

The number of social networking users in Europe is growing fast, with most of them unaware of the privacy and security consequences of the personal data they make available online. Are we in dire need of new and improved privacy laws? Should the companies running social networking sites make sure their users understand the privacy implications of their actions even though it hurts their bottom line?

The European Commission is adopting new data protection rules that will strengthen the position of citizens as well as consolidating existing data protection requirements into one new single EU law. The new rules will also require data controllers to make data protection integral to their processes. ENISA was one of the bodies consulted before the new directive was produced, and the new rules will give a sound legal framework for protecting privacy.

However, implementation will be challenging. Service providers and all other data controllers will need to fully understand and comply with their responsibilities.

With regard to social networks in particular, users need to be aware of what information they are sharing and who may be able to access it, now or in the future. Social network providers certainly have a role to play in ensuring that users understand privacy, and how information will be shared.

As for the service providers' bottom line, we'd hope that users, and therefore advertisers, will go to the sites they know will respect their privacy and protect their personal data.

Of course, there are always risks from deliberate abuse of social media sites. For example, an investigation earlier this year in the UK found that more than 80 children were groomed for sexual abuse through the online game Habbo Hotel. This happened even though the company had signed up to the European Commission's Safer Social Networking Principles. For all users, and particularly children, service providers need to show that they are fully complying with their privacy and security responsibilities. The alternative is further regulation, which could limit freedom and economic opportunity, and in any event, may prove unworkable in practical terms.

What are your future plans for ENISA? What would you like to focus on in more detail?

We've often said that no one state or organization has all the answers when it comes to ensuring cyber security. Our plans for the future include a lot more collaboration, and building bridges between the diverse groups and individuals involved in cyber security, so that we can find answers together. For example, we will be cooperating very closely with Europol on its new Cybercrime Centre in The Hague, looking particularly at security and crime prevention.

ENISA also has an excellent reputation as an information broker, and this is also something we plan to build on by helping all of our stakeholders to share and learn from each other. In addition to this facilitator role, ENISA also acts as a centre of expertise on network and information security.

One of our work areas looks at how we can assess and be prepared for emerging and future risks, and this is an area that we plan to develop further.

Cyber attackers are becoming more sophisticated in their approaches, as we've seen recently with the Flamer spyware attacks in the Middle East, and the Stuxnet worm before that, which targeted control systems.

If we can look ahead, to predict what types of attack are being planned and how they might be launched, we can stay one step ahead of the cyber criminals and terrorists. Of course, ENISA itself will not have all of the answers, but by working with all of our stakeholders, we can ensure that Europe's citizens and economy have the highest possible levels of network and information security.

Mirko Zorz is the Editor in Chief of (IN)SECURE Magazine and Help Net Security.

Unintended, malicious and evil applications of augmented reality

by Gregory Conti, Edward Sobiesk, Paul Anderson, Steven Billington, Alex Farmer, Cory Kirk, Patrick Shaffer, and Kyle Stammer

Most new products begin life with a marketing pitch that extols the product's virtues. A similarly optimistic property holds in user-centered design, where most books and classes take for granted that interface designers are out to help the user. Users themselves are assumed to be good natured, upstanding citizens somewhere out of the Leave it to Beaver universe.

In reality, however, the opposite is often true. Products have substantial flaws, technology designers seek ways to extract money from users, and many users twist well-intentioned technology in ways the designers never expected, often involving baser instincts.

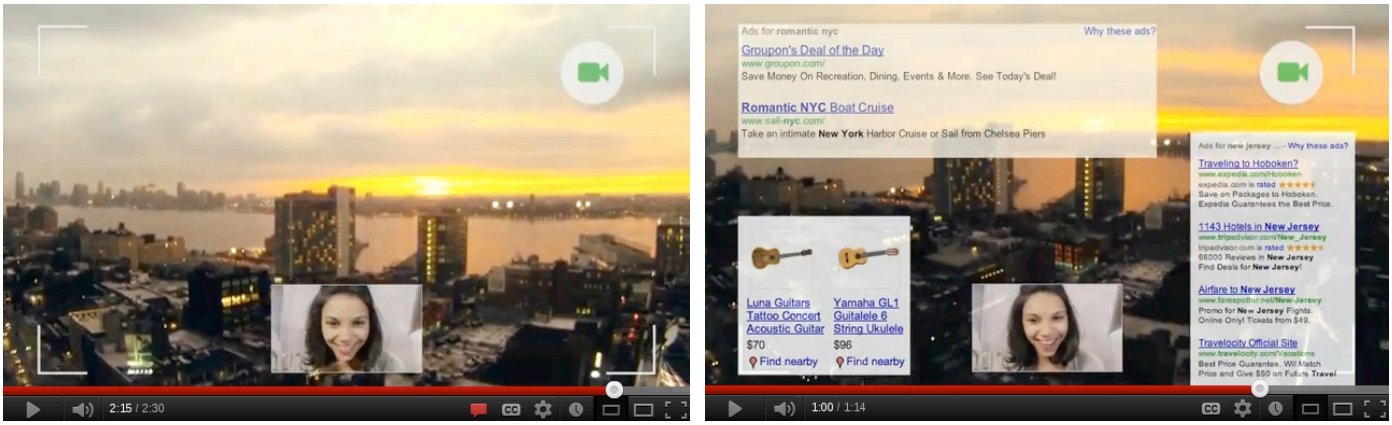
These realities should come as no surprise to security professionals who are usually most effective when assuming the worst of people. One sure to be abused emerging technology is augmented reality.

Augmented reality technologies overlay computer generated data on a live view of the real world.

Anticipated application domains include entertainment, travel, education, collaboration, and law enforcement, among numerous others.

Augmented reality bears great promise as exemplified by Google's highly optimistic "Project Glass: One day..." video. In the video, a theoretical descendent of Google's Project Glass helps the user navigate a city, communicate, learn the weather, and otherwise manage his day.

A day after Google posted the video, YouTube user rebelliouspixels posted a parody video "ADmented Reality" that remixed Google's Project Glass vision with Google Ads. As we look to the future, this less optimistic view likely will be closer to the mark.



Optimistic view of future augmented reality as envisioned by Google (left). A more pragmatic, and advertisement laden, view by YouTube user rebelliouspixels (right).

In this article, we combine augmented reality with reasonable assumptions of technological advancement, business incentives, and human nature to present less optimistic - but more probable - future augmented reality applications.

Admittedly, some are dystopian. We end with suggestions for the security and usability communities to consider now - so that we may be better prepared for our future of augmented reality and the threats and opportunities it presents.

We do not intend to propose science fiction, but instead consider technologies available today or likely to arrive in the next five to ten years.

Unless otherwise stated, we assume the capabilities and overall popularity of today's iPhone/iPad - always on networking, high resolution video cameras, microphones, audio, voice recognition, location awareness, ability to run third-party applications, and processing support from back-end cloud services - but resident in a lightweight set of eyewear with an integrated heads-up display.

Learning from the past

As we consider potential misuse and risks associated with augmented reality we can learn a great deal from past desktop applications and current iPhone and Android apps to gain insight into both human nature and technical possibilities. From this analysis we identify at least three primary threat categories.

The first category is simplest, current applications that are easily ported to future systems, with little to no augmentation.

The next category includes hybrid threats that are likely to evolve due to enhanced capabilities provided by augmented reality.

The final category, and the hardest to predict, are entirely new applications which have little similarity to current applications. These threats will lean heavily on new capabilities and have the potential to revolutionize misuse.

In particular, these applications will spring from widespread use, always on sensing, high speed network connectivity to cloud based data sources and, perhaps most importantly, the integration of an ever present heads-up display that current cell phones and tablets lack.

Regardless to which category the new threats belong, we assume that human nature and its puerile and baser aspects will remain constant, acting as a driving force for the inception of numerous malicious or inappropriate applications.

Applications

This section lists potential misuse applications for augmented reality. Of course, we do not mean to imply that Google or any other company would endorse or support these applications, but such applications will likely be in our augmented future nonetheless.

Persistent cyber bullying

In the world defined by Google Glasses users are given unparalleled customizability of digital information overlaid on top of the physical environment. Through these glasses this information gains an anchor into the physical space and allows associations that other individuals can also view, share, vote on, and interact with just as they would via comments on YouTube, Facebook, or restaurant review sites.

Persistent virtual tagging opens up the possibility of graffiti or digital art overlaid upon physical objects, but only seen through the glasses. However, hateful or hurtful information could just as easily be shared among groups (imagine what the local fraternity could come up with) or widely published to greater audiences just as it can today, but gains an increasing degree of severity when labeling becomes a persistent part of physical interactions.

Imagine comments like “Probably on her period” or “Her husband is cheating” being part of what appears above your head or in a friend’s glasses without your knowledge. Such abuse isn’t limited to adult users.

The propensity for middle and high school age youth to play games that embarrass others is something to be expected. The bright future predicted by Google may be tainted by virtual “kick me” signs on the backs of others which float behind them in the digital realm.

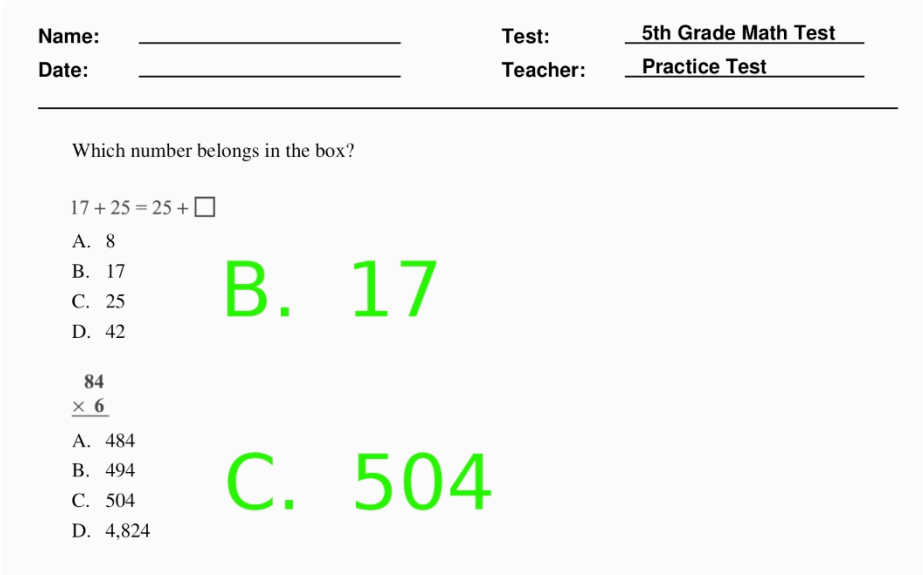
Lie detection and assisted lying

Augmented reality glasses will likely include lie detection applications that monitor people and look for common signs of deception. According to research by Frank Enos of Columbia University, the average person performs worse than chance at detecting lies based on speech patterns and automated systems perform better than chance. Augmented reality can exploit this. The glasses could conduct voice stress analysis and detect micro-expressions in the target’s face such as eye dilation or blushing.

Micro-expressions are very fleeting, occurring in 1/15 of a second, beyond the capabilities of human perception. However, augmented reality systems could detect these fleeting expressions and help determine those attempting to hide the truth. An implication is that people who use this application will become aware of most lies told to them. It could also provide a market for applications that help a person lie.

Cheating

Gamblers, students, and everyday people will likely use augmented reality to gain an unfair advantage in games of chance or tests of skill. Gamblers could have augmented reality applications that will count cards, assist in following the “money card” in Three Card Monte, or provide real-time odds assessments. Students could use future cheating applications to look at exam questions and immediately see the answers.



Future augmented reality applications will likely assist cheating. In this notional example the student sees the answers by simply looking at the test.

Stealing

Theft and other related crimes may also be facilitated by augmented reality. For example, persistent tagging and change detection could be used to identify homes where the occupants are away on vacation. We anticipate augmented reality will perform at levels above human perception. Applications could notice unlocked cars or windows and alert the potential criminal.

When faced with a new type of security system, the application could suggest techniques to bypass the device, a perverted twist on workplace training. The Google Glass video depicted the user calling up a map to find a desired section of a book store. We anticipate similar applications that might provide escape routes and locations of surveillance cameras.

Law enforcement detection

We also anticipate other applications to support law breaking activities. Today's radar and laser detectors may feed data into drivers' glasses as well as collaboratively generated data provided by other drivers about locations of traffic cameras and speed traps. Newer sensors, such as thermal imaging, may allow drivers to see police cars hidden in the bushes a mile down the road. License plate readers and other machine vision approaches will help unmask undercover police cars. Counter law enforcement applications will certainly move beyond just driving applications and may assist in recognizing undercover or off duty police officers, or even people in witness protection programs.

Front and rear looking cameras would allow users to see behind them and collaborative or illicit sharing of video feeds would allow users to see around corners and behind walls. Average citizens may use their glasses to record encounters with police, both good and bad.

Law enforcement

Law enforcement variants of augmented reality may dramatically change the interaction between police officers and citizens. The civil liberties we enjoy today, such as freedom of speech and protection against self-incrimination, will certainly be affected by impending augmented reality technology. What might be relatively private today (such as our identity, current location, or recent activity) will be much more difficult to keep private in a world filled with devices like Google Glasses.

A key enabler of future augmented reality systems is facial recognition. Currently, facial recognition technology is in a developmental stage, and only established at national borders or other areas of high security.

Ralph Gross, a researcher at the Carnegie Mellon Robotics Institute, claims that current facial recognition technology is becoming more capable of recognizing frontal faces, but struggles with profile recognition. Current technology also has problems recognizing faces in poor lighting and low resolution.

However, we anticipate significant advances during the next decade. Law enforcement agencies, like the police department in Tampa, Florida, have tested facial recognition monitors in areas with higher crime rates, with limited success. The primary cause behind these failures has been the inability to capture a frontal, well lit, high resolution image of the subject. This obstacle blocking effective facial recognition would be quickly removed in a world where augmented reality glasses are common and facial images are constantly being captured in everyday interactions.

While facial recognition via augmented reality (through glasses or mobile devices) might seem harmless at first glance, a deeper look into this new technology reveals important unintended consequences. For example, a new form of profiling may emerge as a police officer wearing augmented reality glasses might recognize individuals with prior criminal records for which the subjects have already served their time. Without augmented reality, that police officer would have likely never recognized the offenders or known of their crimes.

Of course augmented reality may be very beneficial to law enforcement activities, but raises serious questions about due process, civil liberties, and privacy. The end result may be a chilling effect on the population as a whole, both guilty and innocent.

Dating and stalking

Augmented reality opens the flood gates to applications for dating and stalking. Having a set of eyeglasses that records and posts your location on social networks means that everybody you know can see where you are. For example, a man sits down at a bar and looks at another women through his glasses, and her Facebook or Google+ page pops up on his screen (since she did not know to limit her privacy settings).

While augmented reality brings vastly new and exciting opportunities, the technology threatens to eliminate the classic way of meeting and getting to know people: by actually spending time with them.

Consider an application that already exists: "Girls Around Me". Girls Around Me uses data from social networking sites to display locations of nearby girls on a map. According to Nick Bilton of The New York Times, this application "definitely wins the prize for too creepy."

The "Girls Around Me" app for smart phones, which uses social networking data to locate nearby women, portends a future of creepy, but plausible augmented reality uses.

The evolution of such applications combined with augmented reality opens up numerous other possibilities. Perhaps the glasses will suggest pick-up lines based on a target's interests, guess people's ages, highlight single women (or married women), make people more attractive (virtual "beer goggles"), or provide "ratings" based on other users' feedback. Lie detection applications will likely be in frequent use, and misuse. Expect continuous innovation in this domain.

Recreational pharma

We anticipate that augmented reality will be used to emulate or enhance drug use. History has taught us recreational drugs will always be in demand as will be additional means of enhancement. Some may recall the combination of drugs with Pink Floyd laser light shows.

Others may have experimented with Maker SHED's Trip Glasses which suggests users "Enjoy the hallucinations as you drift into deep meditation, ponder your inner world, and then come out after the 14-minute program feeling fabulous" or the audio approaches suggested by Brad Smith's DEFCON 18 "Weaponizing Lady GaGa" talk. Augmented reality will open up significant and sought after possibilities.

Erotica

Let's face it, porn is a driving force behind Internet and technological growth, and we believe the same will hold true for augmented reality.

Augmented reality will facilitate sexual activities in untold ways including virtual sexual liaisons, both physical and virtual, local and at a distance.

Advanced sensors may allow penetration of clothing or the overlay of exceptionally endowed features on individuals in the real world, perhaps without their knowledge. The advice frequently given in public speaking classes, "Imagine the audience naked," takes on entirely new meaning in this era.

Surveillance

There are more than 300 million people in the United States alone and more than that number of mobile phones. Imagine if even one third of this group actively wore and used augmented reality glasses. That would mean 100 million always-on cameras and microphones wielded by adults, teenagers, and children continually feeding data to cloud-based processors.

Virtually no aspect of day-to-day life will be exempt from the all seeing eye of ubiquitous and crowdsourced surveillance. Businesses will be incentivized to collect, retain, and mine these data flows to support business objectives, such as targeted advertising, and governments will covet and seek access to this data for security and law enforcement aims.

The implications of the privacy of the individual citizen and the chilling effect on society as a whole could be profound.

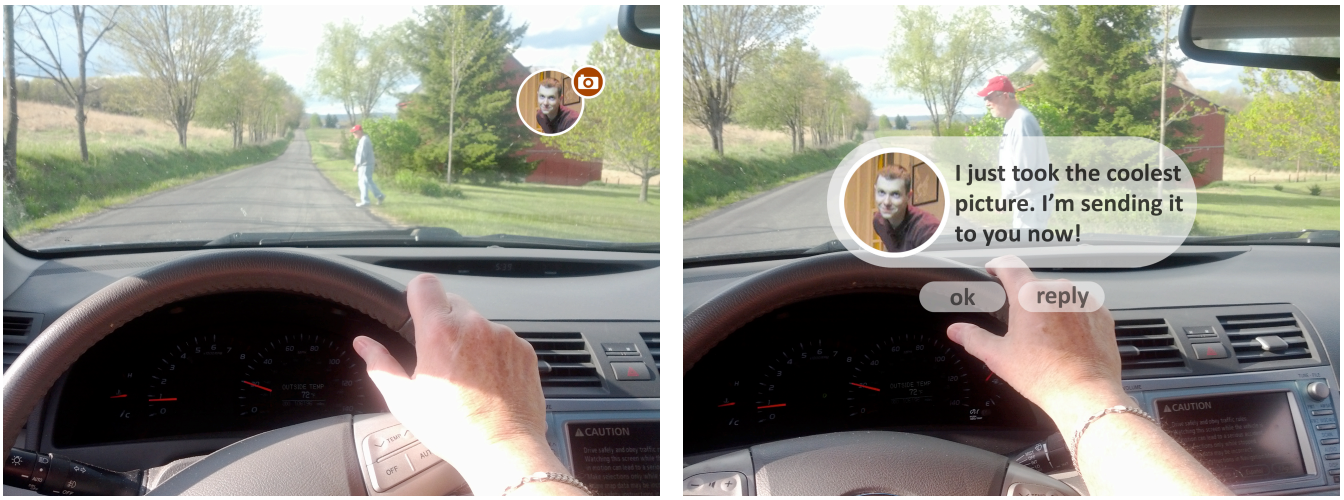
Distraction

People have long been concerned about the danger of billboards when driving, because they take drivers' eyes off the road. Text messaging while driving is widely illegal because of the distraction it causes.

Now consider augmented reality glasses with pop-up messages that appear while a person drives, walks across a busy intersection, or performs some other activity requiring their full attention.

For anybody wearing the glasses, text messaging or advertising alerts and similar interruptions would be very distracting and dangerous. You've likely seen, on many occasions, drivers attempting to use their cell phones and their resultant erratic driving.

Augmented reality devices encourage such "multitasking" behavior at inappropriate times. The results will not be pretty. Consider the example below of a driver reading a text message while a pedestrian is crossing the road.



Driver wearing augmented reality glasses receives text message and is too distracted to notice a pedestrian crossing street.

Voyeurism

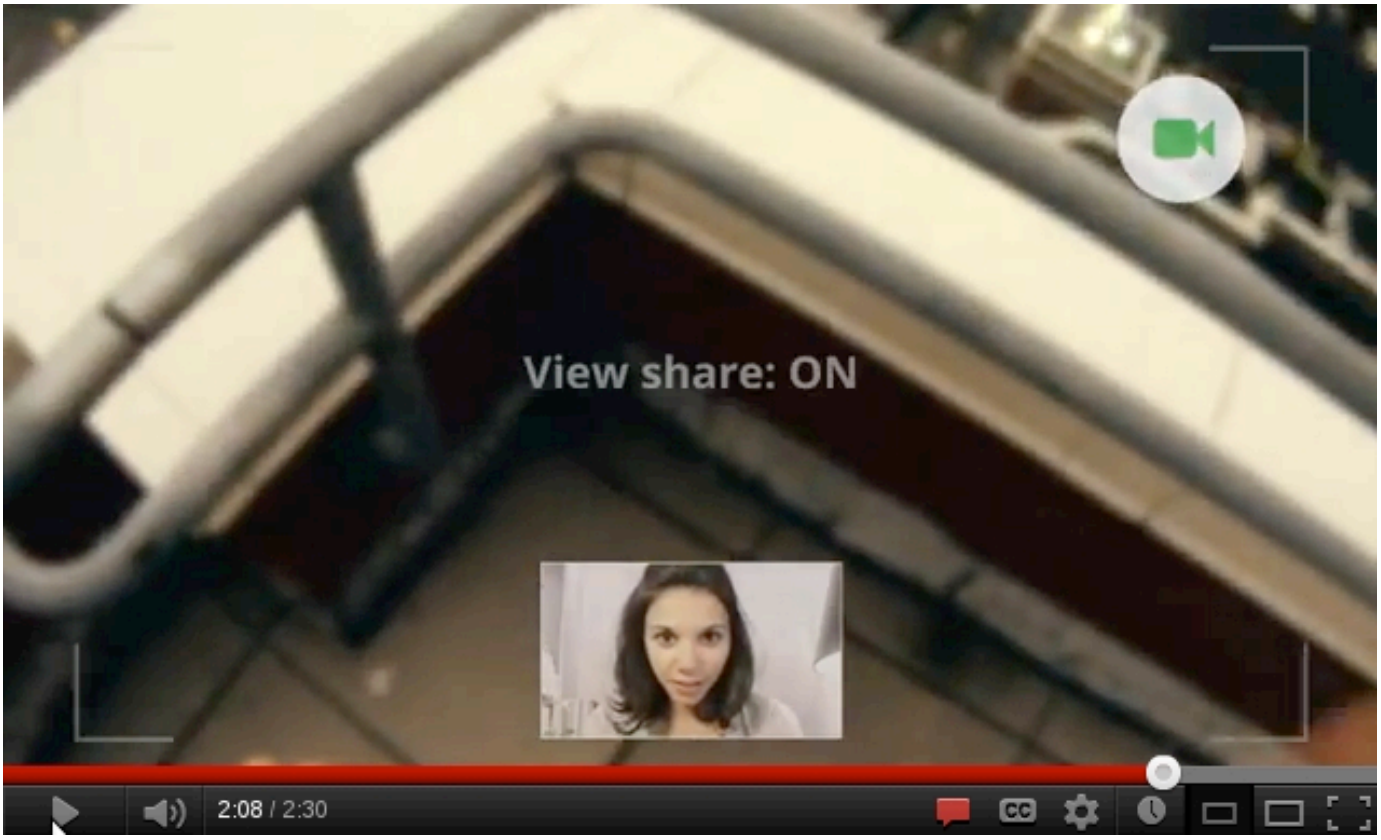
People today do stupid things (see the movie Jackass for textbook examples), and in the future, people will continue to do stupid things while wearing augmented reality glasses. One commenter on Google's YouTube video, PriorityOfVengence1, suggested that someone might even commit suicide wearing Google Glasses.

Man: Hey, wanna see something cool?
Girl: Sure!
Man jumps off building

The context of this comment refers to the end of the video when the main character is on a roof video chatting with his girlfriend and says "Wanna see something cool?"

PriorityOfVengence1's comment received over sixty thumbs up in just three days. While some might laugh at the comment, it highlights a disturbing potential reality.

What if people spiraling into depression began streaming their suicide attempts by way of their glasses? It is certainly possible - this and many other variations of augmented reality voyeurism should be anticipated.



In the Google Glasses video the main character stands near the edge of a balcony in a live video chat with his girlfriend. One YouTube commenter suggested Google Glasses might be worn while attempting suicide.

Untrusted reality

The focus of this article is on user applications that behave in accordance with the user's wishes. However, if we expand our assumptions to allow for malicious software, options become even more interesting. With malicious software on the augmented reality device, we lose all trust in the "reality" that it presents.

The possibilities are legion, so we will only suggest a few. The glasses could appear to be off, but are actually sharing a live video and audio feed. An oncoming car could be made to disappear while the user is crossing the street. False data could be projected over users' heads, such as a spoofed facial recognition match from a sexual offender registry.

For related malware research on today's mobile technology see Percoco and Papathanasiou's "This is not the droid you're looking for..." from DEFCON 18 to begin envisioning additional possibilities.

Conclusions

The era of ubiquitous augmented reality is rapidly approaching and with it amazing potential and unprecedented risk. The baser side of human nature is unlikely to change nor the profit oriented incentives of industry. Expect the wondrous, the compelling, and the creepy. We will see all three.

However, we shouldn't have to abdicate our citizenship in the 21st century and live in a cabin in Montana to avoid the risks augmented reality poses.

As security professionals we must go into this era with eyes wide open, take the time to understand the technology our tribe is building, and start considering the implications to our personal and professional lives before augmented reality is fully upon us. To live in the 21st century today online access, social networking presence, and instant connectivity are near necessities.

The time may come when always on augmented reality systems such as Google Glasses are a necessity to function in society; before that time however we must get ahead of the coming problems. The first few kids who walk into their SAT exams wearing augmented reality glasses and literally see the answers are going to open Pandora's Box.

Gregory Conti is Director of West Point's Cyber Research Center and an Associate Professor of Computer Science. He is an active researcher in usable security, security data visualization, online privacy, and cyber warfare.

Edward Sobiesk is Director of West Point's Information Technology Program and an Associate Professor of Computer Science. His research interests include electronic privacy, usable security, and computing education.

Paul Anderson, Alex Farmer, Patrick Shaffer, and Kyle Stammer are recent graduates of West Point where they studied computer science and information technology.

Steven Billington and Cory Kirk are currently seniors at West Point where they are studying computer science and information technology.

The views in this article are the authors' and do not reflect the official policy or position of the US Military Academy, the Department of the Army, the Department of Defense, or the US Government.

SANS Forensics Prague 2012

Prague, Czech Republic | 7-13 October 2012

THE Seven-Day Digital Forensics Experience!

Including:

Four Top-Level Digital Forensics Courses

- FOR408: Computer Forensic Investigations - Windows In-Depth
- FOR508: Advanced Computer Forensic Analysis and Incident Response
- FOR563: Mobile Device Forensics
- FOR610: Reverse-Engineering Malware: Malware Analysis Tools & Techniques

European Debut of Mobile Device Forensics

Industry Expert Speakers

Featuring:

European Digital Forensics and Incident Response Summit

7 October 2012

Do not miss out on this unique opportunity, places are strictly limited so register now www.sans.org/forensics-prague-2012

5% discount to readers of (In)Secure Magazine

Enter the code **SANSFP12ISM5** when registering

Microsoft's study into unsecure supply chains leads to botnet disruption

Microsoft's Digital Crimes Unit has disrupted the functioning of yet another botnet by effecting a takedown of a domain which was also hosting over 500 different strains of malware and has been linked to malicious activity since 2008.

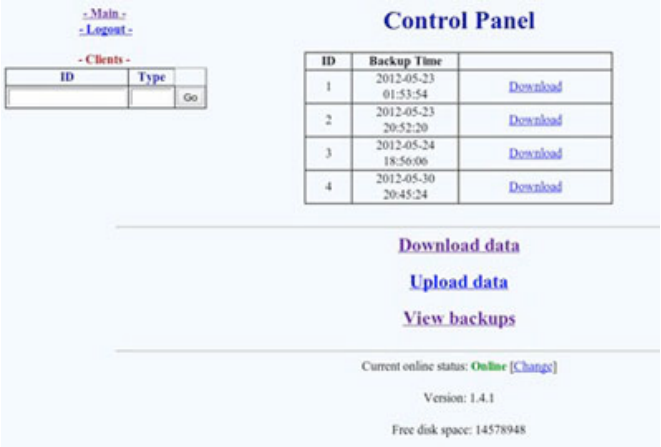
The takedown is the result of an investigation that Microsoft has launched in order to find out how criminals use supply chains to introduce counterfeit software embedded with malware. During the investigation they discovered that twenty percent of the PCs the researchers bought from an unsecure supply chain were infected with malware, and that unsuspecting victims have their newly bought computers automatically enslaved in a botnet and ready to spy on its owners and infect other computers by spreading via USB flash drives.

By filing suit with the U.S. District Court for the Eastern District of Virginia, Microsoft was granted a restraining order against an individual by the name of Peng Yong, his company, and three other unnamed individuals behind the scheme, as well as the permission to transfer the hosting of the domain in question (3322.org) and nearly 70,000 of its subdomains to Microsoft.

The malware strains found on these subdomains are capable of many malicious actions - from remotely turning on an infected computer's microphone and video camera to recording keystrokes.

"The Nitol botnet malware itself carries out distributed denial of service (DDoS) attacks that are able to cripple large networks by overloading them with Internet traffic, and creates hidden access points on the victim's computer to allow even more malware - or anything else for that matter - to be loaded onto an infected computer," explained Richard Boscovich, Assistant General Counsel with the Microsoft Digital Crimes Unit.

Analysis of Flame C&C servers reveals more unknown malware



Kaspersky Lab and Symantec, in conjunction with ITU-IMPACT and CERT-Bund/BSI, have revealed worrisome new discoveries about other malware that seems to have been created and used alongside Flame, after having analyzed two of the C&C servers and the information found on them.

The servers could be accessed through a Web application called Newsforyou, which processes the W32.Flamer client interactions and provides a simple control panel - so simple, in fact, that it could be mistaken for a content management system for a blog or a news outlet.

"The C&C developers didn't use professional terms such as bot, botnet, infection, malware-command or anything related in their control panel. Instead they used common words like data, upload, download, client, news, blog, ads, backup etc," shared Kaspersky Lab experts. "We believe this was deliberately done to deceive hosting company sys-admins who might run unexpected checks."

But the most important discovery is the fact that the application for the control panel hasn't been exclusively used for Flame. "It contains functionality that allows it to communicate with computers compromised with multiple malware identifiers using different protocols," the researchers say.

There are four active protocols, and only one is used by Flame. The malware using the remaining four is unknown - could be Flame variants, or totally different malware altogether. But according to Kaspersky researchers, one of these Flame-related unknown malicious objects is currently operating in the wild.

"The servers were set up to record minimal amounts of information in case of discovery. The systems were configured to disable any unnecessary logging events and entries in the database were deleted at regular intervals. Existing log files were securely deleted from the server on a regular basis.

These steps were taken in order to hamper any investigation should the server be acquired by third parties," points out Symantec.

"The attackers were not thorough enough, however, as a file revealing the entire history of the server's setup was available. In addition, a limited set of encrypted records in the database revealed that compromised computers had been connecting from the Middle East.

We were also able to recover the nicknames of four authors—D***, H*****, O******, and R***—who had worked on the code at various stages and on differing aspects of the project, which appear to have been written as far back as 2006."

The thing that seems to confirm the theory that the people behind this were not well-funded criminals, but were part of a military and/or intelligence operation, is that the server operators could not know which modules were pushed out to which machines because the control panel does not function as transparently as most other ones, and the collected information that was stolen from compromised computers was stored on the servers but in encrypted format, and no key to decrypt it was found on it.

Blackhole 2.0 is out with new exploits and same price

A new version of BlackHole, one of the most popular exploit kits out there, has been made available by its creator, who has supposedly rewritten it from scratch.

BlackHole 2.0 brings many improvements:
• Dynamic URL generation in order to foil the automatic systems for downloading exploits used by security researchers
• The removal of exploits for "old" vulnerabilities, and the inclusion of three different exploit packs - one including Java exploits, the second exploits for the Adobe PDF LibTiff vulnerability (CVE-2010-0188), and the third for Internet Explorer's Microsoft Data Access Components flaw (CVE-2006-5559) - a rather old vulnerability that still gets taken advantage of because of unpatched IE6 browsers
• Links can get renamed to human readable format (for example /news/index.php) instead of kept in the obviously suspicious format that includes a slew of random characters
• JAR and PDF exploits run only if vulnerable versions of plug-ins are detected, so they don't trigger detection by antivirus package unnecessarily
• A new administration panel with a considerable number of new options.

In spite of all these changes, the new version of the exploit kit costs the same as the previous one: a one-year license amounts to $1500.

Shamoon attacks persist

While it still unknown whether the recent attacks against Saudi Aramco and RasGas were part of the so-called Shamoon attacks, the latter are continuing unabated, says Symantec. These newest attacks also use a more recent variant of the destructive Disstrack malware.

Initially, the malware would drop a wiper component and it would first wipe a prioritized list of files contained in the Documents and Settings, Users and System32\Config folders by overwriting them with a 192KB block filled with a partial JPEG image of a burning United States flag, then the computer's Master Boot Record and its active partition.

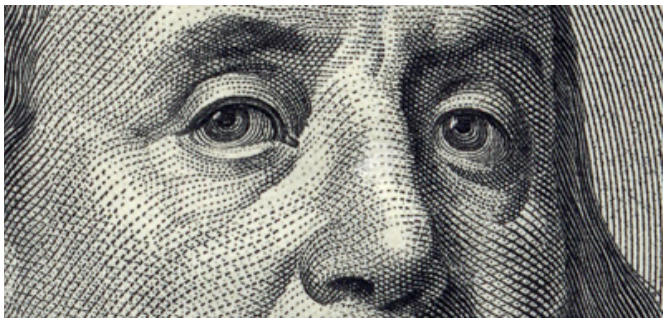
This new variant isn't into making a statement, so the 192KB block that overwrites the files contains only randomly generated data.

Unfortunately, the initial infection vector has still not been confirmed, so it's difficult to say what likely targets should be on the lookout for.

The malware can be detected by a variety of desktop AV solutions, but if you don't have one, checking for and finding a service called ddr, a file called ddr.sys in the %System% \Drivers folder and ddrisk.sys in the %System %\Drives folder may indicate that your machine has been compromised.

Still, this is a problem that individual users will likely not be faced with, as the Shamoon attacks have been very limited and extremely targeted.

Mobile malware has become a profitable industry



Lookout released its State of Mobile Security Report 2012 which explains the issues that individuals faced on mobile devices this year and explores the prominent trends in mobile threats.

Mobile malware has now become a profitable industry

Because of its global ubiquity as a phone payment mechanism, premium text billing is the most common tactic used by malware writers to commit financial fraud on mobile. This class of malware, termed “Toll Fraud,” has become the most prevalent type of malware within the past year.

Just one family of Toll Fraud malware, FakeInst, accounted for 82 percent of Lookout user detections in June 2012 and is estimated to have successfully stolen millions of dollars from people in Russia, the Middle East, and parts of Europe.

Mobile privacy is a growing issue

Privacy is one of the biggest issues people face on mobile devices. In 2012, a significant portion of privacy problems arose from aggressive advertising techniques, including pushing out-of-app ads and accessing personally identifiable information without user notification.

Lookout estimates that five percent of Android applications include these aggressive ad networks and these apps have been downloaded more than 80 million times.

Geography and user behavior are main drivers for encountering threats

People in Russia, Ukraine and China have a significantly higher likelihood of encountering malware than elsewhere. User behavior is the other leading factor; people who download apps outside of a trusted source, like Google Play, have a higher chance of encountering malware.

Visiting unsafe links from a mobile device is one of the most common ways people encounter mobile threats.

Web-based threats like phishing are often able to target both traditional PC users and mobile users equally, making these schemes easy for malware writers to produce and replicate.

Lookout’s detected that four out of ten mobile users click on an unsafe link over the course of a year.

Gaming the app ecosystem

Lookout observed malware designed to enable shady app promoters to conduct download fraud. These malware families primarily affected users in China.

In the past year, Lookout discovered malware capable of automatically downloading apps from alternative app market sources without the user's knowledge, rooting the phone to download additional apps without warning, or installing third-party app stores.

FinFisher commercial spyware toolkit goes mobile

The existence of FinFisher, a commercial spyware toolkit created by UK-based Gamma Group International, has recently grabbed the attention of the general public when two security researchers from Toronto released the results of the analysis of FinSpy, a module that is part of the toolkit and gets installed on PCs.

The samples for the analysis were provided by two pro-democracy Bahraini activists who received them via faked emails, and the analysis revealed that FinSpy is a very thorough spying tool that is capable of recording chats, screenshots, keystrokes, grabbing other information from infected systems and passing it on to C&C servers set up by the attackers around the world.

Following this discovery, Gamma Group stated that they did not sell any of their products to Bahrain, and that the sample the researchers received was probably stolen or a result of reverse-engineering efforts.

Now those same researchers - Citizen Lab security researcher Morgan Marquis-Boire and Berkley computer science doctoral candidate Bill Marczak - have received samples that proved that FinFisher also has a component that can spy on mobile users.

Called FinSpy Mobile, the spyware records calls, text messages, emails, downloaded files, keystrokes and audio sounds via the devices' microphone, makes silent calls, extracts contact lists and uses GPS to keep tabs on the users' position.

No mobile user is safe, it seems, as FinSpy Mobile is able to compromise iOS, Android, BlackBerry, Windows Mobile and Symbian-run devices.

Still, the component can't be installed without user interaction, and the researchers speculate that the targets get infected via socially engineered e-mails, Trojanized apps, or even by someone they know who downloads and installs the malware without the user knowing.

"Win 8 Security System" rogue AV spotted

Windows 8 has not yet been released and cyber crooks are already taking advantage of its name.

McAfee researchers have recently spotted a new rogue AV solution dubbed “Windows 8 Security System” which, at first glance, does look rather legitimate.

"Win 8 Security System will display lots of fake alerts and messages and will show a scan window on each system boot. It will display lots of detections, though it is obvious these are fake," the researchers warn.
But even if the victims realize that the software in question is a fake and aimed only at bilking users of their hard-earned cash, they will have a tough time with removing it.

To protect itself, the malware comes with a rootkit and creates a bucketload of registry elements and values, as well as half a dozen files and one folder, making it almost impossible to manually remove as you can permanently damage your system if you make any mistakes in the process.

The researchers recommend manual removal only to experienced users such as IT specialists or highly qualified system administrators. Other users should use their regular desktop security software.

The enemy at the gate

by Gavin Watson

Several high profile breaches have highlighted the important role that social engineering plays in targeted attacks on companies. As a result, more organizations are requesting social engineering assessments as part of their pen tests. This article outlines some of the most serious security issues that are regularly highlighted and how they can be remediated.

Sensitive document metadata

A social engineer will undertake meticulous preparation before setting foot on your premises. If your company website hosts documents, you may be revealing sensitive information.

Documents saved in various Microsoft software packages such as Word, Excel and Powerpoint and Adobe Acrobat can contain metadata that can potentially be used in attacks against your company.

The metadata may contain the name of the document author, their username, company name, the path where the document was saved, the version of the software that produced it and even file share paths. The collected usernames alone can be used to enumerate targets for phishing attacks and the version of the software will help to refine the exploits used to compromise the host.

Attackers can collect this information using freely available tools such as FOCA and Metagoofil. Fortunately, metadata is straightforward to remove and various step-by-step guides are available online.

Email authentication

When attempting to reset a remote administration or email account password, companies will often request that a manager sends an email to the operator to confirm the reset.

There are free online services such as ‘hoaxMail’ that aid in sending spoofed emails. The bottom line is that emails should never be used to validate an individual.

Mixing personal and business social media accounts

Social media websites are an extremely rich attack vector. Fake 'puppet profiles' are used by social engineers to establish connections with individuals and obtain information that is used to help gain credibility when telephoning through to the company.

Generic staff badges and lanyards

One of the most effective props a social engineer can have is a realistic looking staff or contractor badge. The social engineer may hang around the staff smoking area or car park and secretly take photos of the badges worn by staff members in order to replicate them. However, simply instructing staff to not show badges means that the attacker doesn't need to do anything, as it is the norm for badges not to be shown.

A good defense is to have branded lanyards and badges that correspond to departments or position within the company. This way the attacker needs to establish what variant of badge and lanyard to copy.

Ineffective challenging

It is obvious that staff should challenge people they don't know who enter the work place. However, a well prepared social engineer will have a full pretext ready and will have pre-empted possible questions.

Therefore, your policy and staff awareness training should stipulate that a challenge needs to be combined with confirmation, such as calling someone to escort them, who should know who the visitor is and why he or she is on the premises.

SOCIAL MEDIA WEBSITES ARE AN EXTREMELY RICH ATTACK VECTOR.

Sensitive information in plain sight

Social engineers with always check for passwords written on Post-It notes under the keyboard and in the drawers. However, aside from this obvious breach of security policy, our assessments regularly show that IP addresses and MAC addresses are found stuck to devices such as switches and routers. Network topology documents and lists of workstation hostnames have been found pinned to cabinet doors.

Companies often display names of staff members on public notice boards, and they can be converted into usernames for attacks against the network. Social engineers use snippets of apparently harmless information to obtain more useful data. Anything that the company does not actively publish should be kept out of plain sight.

Live network ports

Social engineering assessments will commonly target server rooms as primary objectives. However, the value of this is questionable if an attacker needs to only plug his laptop into a network access point in a meeting room.

Alternatively, he could simply sniff network traffic from the car park using a wireless access point or a 3G enabled device used in the vicinity of your office building.

Devices that are extremely effective at launching remote attacks, such as the Pwnie Express and Raspberry Pi with pen-testing software, are available online.

Over reliance on NAC

While network access control (NAC) technologies can help to thwart this type of attack, a social engineer could potentially bypass NAC by first visiting your premises and writing down the MAC address displayed on the back of one of your multifunction printers, then using this to spoof the MAC address of their chosen hacking device.

The only way to reduce this risk is to fully disable unused ports.

Endpoint 2-factor authentication

Assessments commonly reveal that staff members are happy to log in to machines for contractors, as long as they provide a plausible reason.

A pretext as simple as just needing to check the internet connectively on a workstation may be all that is required to trick a staff member into opening up access to your network.

Company policies and staff training should reinforce the correct use of log on credentials and token devices.

Cheap physical locks

If all else fails and a malicious individual has gained entrance to your premises, then strong physical security can help to thwart an attack in progress.

Therefore, social engineering assessments may include attempts to pick the locks on doors, drawers, cabinets and padlocks used to physically secure sensitive data. Lock picking tools can be purchased cheaply and video tutorials can be found on websites such as YouTube. Locks are classed as a delaying mechanism in your defense in depth strategy and should not be relied upon to protect critical assets.

However, our assessments often reveal that wafer locks (one of the easiest to bypass) are used on cabinets containing sensitive files, keys in key boxes, security cameras, alarm boxes, lift panels and office drawers (to name a few). It needs to be understood that cheap locks provide little if any security.

Conclusion

In our assessments we have found numerous examples of companies inadvertently displaying information online and on their premises. When pieced together by a skilled social engineer, this information could form the basis of a broader attack on your most valuable data and that of other organizations that you deal with.

It is important that we change our perspective on what constitutes “sensitive” information and adjust our security policies and best practices accordingly.

Gavin Watson is a senior security engineer and head of the social engineering team at security and compliance company, RandomStorm (www.randomstorm.com). Gavin is an expert lock picker and penetration tester. Using physical and social engineering techniques, Gavin and his team regularly breach the defenses of client companies to demonstrate how their security can be improved.

Top five hurdles to security and compliance in industrial control systems

by Jacob Kitchel

For many decades, Industrial Control Systems (ICS) have been the operational systems relied upon to safely and reliably deliver the essentials of daily life. Sometimes referred to as a Critical Infrastructure, they are the backbone of a modern economy. With these systems generally working well, there has been little need to make major changes to them. There has been innovation and some incremental changes, but in the ICS world, it has largely been 'business as usual.'

That's very different than other industries and sectors, such as enterprise IT, where seismic technology shifts seem to occur about every two years. Change in industrial control environments has been handled at a more measured pace and with a lot more caution.

There are several good reasons for this. The first is that the processes these systems control are usually very large and critical to the general public and the normal functioning of society. They support the provisioning of essentials like electricity, water, oil and gas and other basics. If these systems go down, people's health and safety are quickly put at stake. For that reason, reliability and availability have long been the overriding priorities in the design and operation of these systems, making broad-based changes in these environments a real challenge. That's why slow, methodical and incremental change has been the norm for so long.

Another reason why ICS and Supervisory Control and Data Acquisition (SCADA) environments have not seen a more rapid rate of change was because it was not needed. Designed for a simpler era, automation systems typically were designed as proprietary (closed) systems and were implemented in isolated settings, both physically and electronically.

For many years, these systems successfully controlled industrial processes without requiring direct connections to enterprise networks, the Internet, or too much else for that matter.

But the time has come to upgrade or replace these aging systems. There are now compelling reasons to connect these systems to corporate networks and the Internet.

As those connections are made, the isolation – or 'air gaps' – that protected these systems disappears. The long-standing strategy of 'security through obscurity' no longer holds up.

In addition, corporate and operations staffs have other realities and requirements to consider, including:

• Shifting from proprietary to open, standards-based solutions can lower costs, increase operational flexibility and avoid vendor lock-in
• Generating real-time business intelligence from operational data can enhance service delivery
• Improving the effectiveness of automation systems drives new efficiencies into the industrial processes they control, yielding better performance and results
• Ensuring that the operational health and safety levels of the systems and processes are continually maintained.

Another major change that ICS and SCADA system professionals must manage is the explosive growth in the number of intelligent endpoints in industrial environments.

In rapidly growing industry segments such as the Smart Grid, the numbers and types of networked and IP-enabled devices is increasing exponentially. This array of issues, including economic, operational and technological drivers, is forcing automations systems professionals to grapple with much more change at a much faster pace than ever before.

The following are five of the major hurdles that critical infrastructure and industrial process companies often face as they move forward with initiatives to modernize their control environments.

Organizations should consider protocol-aware gateways or firewalls to restrict access and add a layer of security, since many industrial protocols lack authentication and security features.

1. Lack of "last mile" coverage and instrumentation for device visibility – ICS systems are increasingly leveraging wireless and Internet connectivity to expand the system's reach and effectiveness. Gaining faster access to more granular and real-time data from far-flung end points can produce substantial operational benefits. From a security perspective, however, such expansion introduces new risks.

One of the primary security issues that arise in these implementation scenarios stems from the fact that embedded devices often lack local or remote logging capabilities. As a result, they cannot adequately log relevant security and compliance data.

Additionally, interactive remote access can be cumbersome, hard to achieve or only available in an insecure manner.

To address the lack of visibility largely inherent in these devices, organizations should place network sensors logically near the devices to detect events which would normally be present in event logs. Network Intrusion Detection Systems and network flow tools are two such examples.

Organizations should consider protocol-aware gateways or firewalls to restrict access and add a layer of security, since many industrial protocols lack authentication and security features.

2. Not so automatic "automation" – Whether or not they have the Critical Infrastructure designation, ICS operators of all kinds face growing internal and external (regulatory) requirements to produce ever-increasing amounts of operational data.

It is a growing operational and administrative burden, and automation systems operators must find an efficient and secure way to deal with it. Since old habits – and cautions – die hard, many asset owners are averse to fully automating their data collection processes.

This reluctance to fully automate data collection often leads operators to conduct partial automation efforts. Examples include scripts being run manually on each individual host, or scripts that can run remotely but have to be initiated manually.

These half-measures are not thorough and are often incomplete.

Operators do have other options for addressing this challenge. There are technologies and solutions available on the market today that enable operators to automate all of their data collections processes safely, securely and effectively. By embracing a fully automated approach to data collection, operators can safely meet their data collection and reporting requirements, while also alleviating many hours of manual work and human error.

It should be noted that automating data collection is not the same as “network scanning.” Automated data collection utilizes built-in, administrative capabilities in the cyber assets and can be performed in a controlled manner, which utilizes very little overhead on the cyber assets. “Network scanning” is associated with network-based port scanning, which when not done carefully, can affect cyber asset availability in some cases.

3. "Dirty" data – Often times, raw output from tools used to collect security and compliance data is all-encompassing and complete. That’s the good news. The bad news is that it usually includes data that requires analysis by the asset owner in order to make determinations of security or compliance state. When raw output is treated as analyzed output, asset owners get an inaccurate picture of the security and compliance state of their assets.

For example, in the upcoming NERC CIP-010-5, asset owners are required to create a baseline of each cyber asset, which includes several categories of information, one of which is “logical accessible network ports.” If an asset owner utilizes raw “netstat” output as a final source of data for compliance, there will potentially be many additional records of data that do not apply, such as records for local host-only services, which are not available as “logical accessible network ports.”

For most ICS and automation system operators, baselining and tracking expected behavior is difficult, and requires lots of time and specialized expertise.

4. Inability to detect anomalous behavior – Zero-day attacks can be devastating to automation systems. They exploit system vulnerabilities that are unknown at the time of the attack, so there is no patch or fix at the ready, and great damage can often result.

One of the most effective ways to protect against these types of attacks is for operators to continually monitor their networks to develop a baseline of normal activity. This baseline is a reference point that can help operators quickly identify the anomalous, attack-related activity they need to guard against.

However, for most ICS and automation system operators, baselining and tracking expected behavior is difficult, and requires lots of time and specialized expertise.

Additionally, not all applications and operating systems are easy to configure in order to log the data required to accurately detect anomalous behavior. Although asset owners can benefit from having logging and monitoring capabilities in their ICS-process specific applications, more often than not these capabilities are geared solely to making improvements in process performance.

By refocusing their use of these systems to include detection of anomalous – and therefore suspicious – network activity, ICS owners can significantly improve the security posture of their systems

5. Collection, analysis, and workflow lifecycle integration – Many organizations stop at the collection step and then label their security and compliance efforts a success. The fact is that data collection is really just the first step.

To be truly successful, an organization must collect, analyze, and then act on the security and compliance data it gathers from its ICS environment. By continually iterating over and acting upon the data, an organization can track and improve its security and compliance efforts over time.

For example, consider an organization that logs failed logons. If no analysis is performed on the failed logon events, the organization will not know if the failures are malicious or if the events are failed logons from a service that is configured to use an expired password.

Another example, from a compliance perspective, is when an organization logs events to meet a compliance requirement. How will the organization know when log data collection fails or if there is a gap in the collection? Without tracking the dates, times and failures of log collection, the organization leaves itself vulnerable to a compliance deficiency.

Without tracking the dates, times and failures of log collection, the organization leaves itself vulnerable to a compliance deficiency.

Conclusion

The scope and pace of technological change now occurring or coming soon to many ICS environments present new risks to automation systems professionals. But as is always the case with change, risks are accompanied by opportunities.

Old approaches to ICS system design and security are becoming increasingly ineffective in the face of major technology trends and business changes that are now impacting operators. Forward-thinking professionals must find effective ways to overcome these new security and operations challenges.

The first step is recognizing that in many areas of ICS security, what worked in the past likely won't work in the future. Teams must explore new options and develop effective business cases for investing the next-generation ICS security technologies.

By embracing the changes that are taking place in the industry, and adopting new solutions to address them, ICS professionals will be able to mitigate risks and capitalize on the terrific opportunities that lie ahead.

Jacob Kitchel is the Senior Manager of Security and Compliance at Industrial Defender (www.industrialdefender.com). He was also one of the lead security researchers involved in Project Basecamp from S4, discovering multiple vulnerabilities in PLC security.

Criminal psychology has taught us that the thing that deters individuals the most from committing a crime is not the harshness of the punishment, but the likelihood of getting caught. If a law or rule can be broken without anyone ever finding out, it will be broken - no matter the consequences. But if attempts to cheat are sure to be detected, people will not attempt to do it, however small the punishment is.

Preventing an incident from happening is always better than investigating its aftermath. But just as you can't put a policeman on every street corner, it is practically impossible to create a perfectly safe computing system. There will always be threats that are simply not worth protecting against, and there will always be attack vectors you have never considered.

On the other hand, if you can record what happens in your infrastructure and reconstruct the sequence of events for any system in any time range, you have the means to track down all misdemeanors. Being monitored has a large deterrent effect on those who are within your jurisdiction, and helps minimize the threat of an inside job.

The way to achieve this is audit logging and event monitoring, as these methods can provide a reliable recording of who did what and when. The goal is to be able to recreate the sequence of events at a later date, and to be able to trust that recreation. In order to do so, your audit log must be:

Comprehensive: Ensure that the logs include all events that might be important, with all the necessary information about each event.

Trustworthy: The information contained in the logs must be real, reliable, and tamper-proof, so that you can trust it and use it as evidence if needed.

Easy to access: You should be able to access and browse the logs of your entire infrastructure in a single place, as well as perform detailed searches.

In this article, I will try to show you how traditional audit logging fails to fulfill these requirements and how you can solve this problem.

The current state of audit logging

The simplest and most widespread tool to create an audit log is the good old syslog infrastructure, built into almost every operating system and used by most applications. There are no universal standards on what or how to log. Most applications and operating systems log basic events (for example, successful and failed login attempts, logouts, major configuration changes, starting/stopping/restarting services and systems), which can form a good basis for a forensics investigation.

Is this enough? You collect logs for a reason: to be able to find specific information later - the cause of a failure, a security incident, a performance issue, or another problem. It's the information you need, not necessarily the raw data.

Another problem is that vendors rarely provide guidelines about how and what to set up for audit logging. There are not even real standards to be followed by vendors on how to format a log message and what to store in them exactly. Of course there are initiations like Common Event Format (CEF) (www.arcsight.com/solutions/solutions-cef/), Intrusion Detection Message Exchange Format (IDMEF) (www.ietf.org/rfc/rfc4765.txt), or more recently, Common Event Expression (CEE) (cee.mitre.org) and Project Lumberjack (fedorahosted.org/lumberjack/), but neither of these is a widespread, all-around solution.

Even the log itself has many different names and incarnations, like logfile, journal, audit file, event, eventlog, messages file, trail, and so on. If you are lucky, all of them contain at least a timestamp, a hostname or IP address, and some descriptive information.

What is logged and what is not

You might assume that vendors properly set up audit logging on their products and there is no need to change or tweak them. But you would be wrong, as default settings are almost never good enough. For example, the logging defaults on a Windows system are set as follows:

Audit account logon events	success, failure
Audit account management	no auditing
Audit directory access settings	no auditing
Audit logon events	no auditing
Audit object access	no auditing
Audit policy change	no auditing
Audit privilege use	no auditing
Audit process tracking	no auditing
Audit system events	no auditing

Enabling the last two settings causes the system to log everything - and is likely to result in system overload and performance problems. Using the default settings causes logging gaps. Without a clear vision of what you would like to achieve with audit logging, it is difficult to properly configure your systems. Logging everything is not efficient and rather costly: for example, performance, storage, and archiving must be sized accordingly. Also, many log analysis solutions (SIEMs) are licensed based on the amount of log messages processed.

Even if you know which events to log and collect, sometimes the default logging tools are not adequate for the task. In such cases you will need to install external solutions to increase the logging capabilities of the system.

For example, on Solaris you have to install and configure the Basic Security Module (BSM) to track file operations. On an Oracle database server there are many different information sources - the TNS listener log, various trace files, Sqlnet logs (server and clients), SYSDBA audit logs, datafiles for deleted data, redo (and archive) logs, SGA (v$sql, etc), Apache access logs, and more. Even if you are an experienced Oracle administrator, figuring out what level of auditing to setup in a certain scenario can be difficult.

The same login event is logged slightly differently and with different details on almost every operating system or device. Even similar or related systems log the same events differently.

Windows 2008 login message:

2012.07.13. 15:15:34
An account was successfully logged on.

Subject:
  Security ID: SYSTEM
  Account Name: SERVERDEMO$
  Account Domain: DEMO
  Logon ID: 0x3e7
  Logon Type: 10
  New Logon:
  Security ID: DEMO\demouser
  Account Name: demouser
  Account Domain: DEMO
  Logon ID: 0x23f38e08
  Logon GUID: {c38d0279-bc49-b4eb-c93c-2e5f68bda748}

Process Information:
  Process ID: 0x1604
  Process Name: C:\Windows\System32\winlogon.exe

Network Information:
  Workstation Name: SERVERDEMO
  Source Network Address: 10.10.30.112
  Source Port: 45327

Detailed Authentication Information:
  Logon Process: User32
  Authentication Package: Negotiate
  Transited Services: -
  Package Name (NTLM only): -
  Key Length: 0

Ubuntu Linux SSH login message:
Jul 13 15:20:58 serverdemo sshd[6476]: Accepted password for demouser from 10.10.30.112 port 43456 ssh2

CISCO ASA login message
Jul 13 2012 15:20:58: ASADEMO IASA-6-109005: Authentication succeeded for user 'demouser' from 10.10.30.112 /0 to 172.16.1.1/0 on interface outside

Embedded systems, smartphones, and gadgets

Embedded systems and appliances like routers, Wi-Fi hotspots, and other networking devices are common in corporate environments. Also, in addition to traditional desktops and laptops, various other devices are increasingly used to access the company infrastructure - just think of tablets, smartphones, ebook readers, or other similar gadgets.

Logging the activities of these devices is still difficult. Most of them have some type of syslog functionality available, but it is often very limited. Since logs are not forwarded anywhere by default, and storage on the device is limited, only a fraction of the events is logged. Also, the events that are forwarded to the central log server are often random. In other cases, the logging of the system is geared towards helping developers in troubleshooting, and is unusable for reporting or auditing.

Transferring the logs of embedded systems to a central log server can also be problematic, since they often support only the unreliable and unencrypted UDP protocol. This means that sensitive information will be sent over the network without knowing whether any of the messages got lost, and can be also accessed and modified en-route by attackers.

Threat models

In theory, audit logging solutions make it possible for an investigator to figure out what has happened and to track down the attacker. However, several things can hinder their use, or make them outright ineffective in certain situations.

The most basic problem is when the attacker flies under the radar, doing things that don't get logged. For example, most audit logs do not record file operations themselves. Therefore, removing a crucial database file cannot be tracked back to a user. To investigate this, you need to correlate login and logout events with the time of the incident.

But even if the acts of file operations are logged, the changes made on files almost never are. For example, it might be standard operation for a system administrator to change the configuration of the firewall to open a new port for a new server in the DMZ, but allowing unrestricted access everywhere from his workstation to make it easier to run that latest MMORPG is not standard operation (according to a survey, 48% of system administrators has admitted to creating exception rules in the firewall for personal purposes, to get around the IT policy - net-security.org/secworld.php?id=11972).

Another problem is that log messages only contain the most important facts about an event. If the attacker knows what these are, it is not really difficult to mask his malicious acts as something completely innocent. For example, if you log only the filename and file size for file transfers, you will never find out the file *log_messages_for_debugging.tar.gz* (size: 60Mb) downloaded from your server actually contains the credit card database of all the customers of your web shop.

Commands have a similar problem: even if the executed commands are logged, it is possible to create scripts or links which have innocent-sounding names but actually do some very nasty things.

Criminals always try to cover their tracks and remove all evidence that can lead to them. This is especially easy if the evidence is in a digital form: it takes about 10 seconds with vi for the root user to remove incriminating lines from the */var/log/messages* logfile if they're stored only locally, or to disable the logging service completely if the logs are forwarded to a central server.

The problem with log messages is that they correlate very weakly: it is hard to detect removed or changed lines without special countermeasures.

And even if logs are forwarded to a central server and the local administrator cannot tamper with them, a long time can pass between an incident and the reviewing of the logs during an audit. This can leave enough time for an insider to modify the logs, either to remove them or to plant false messages. Again, if log files are stored in a plaintext format, it is trivial to manipulate them without being detected.

What is privileged access monitoring?

Privileged users (administrators, contracted IT providers, executives, and so on) have wide-ranging or even unrestricted access rights to a company's IT assets. By having superuser privileges on servers, these users have the possibility to directly access and manipulate the sensitive information of a company, such as financial or CRM data, personnel records or credit card numbers. Administrators often use shared accounts (over 40% of them: net-security.org/secworld.php?id=11972), making it difficult to track the actions of the users, and to provide proof of any misuse.

Even if the problem of shared accounts can be solved on your servers, many networking devices and appliances have only one single user account.

Privileged Access Monitoring (PAM) tools solve these problems by introducing an independent auditor layer to oversee the working sessions of privileged users. Practically, PAM tools aim to address the following requirements:

1. Managing and controlling privileged sessions (for example, restricting administrative access to the servers)

2. Controlling the users' access to privileged accounts (authenticating the users, restricting access based on time policies, restricting access to system resources)

3. Monitoring access to shared accounts (for example, root or administrator)

4. Collecting audit information for forensics situations, compliance reports, and so on.

Privileged Access Monitoring is still a niche market, with a small but increasing number of commercial players on the field. Since there are number of different ways to approach to the problem, let's review the technology they use:

1. Jump hosts (Hop gateways) provide a web-based interface for accessing the servers: the users access the jump host from their browser, and connect to the target server using a web-based client application that is running on the jump host. In the meantime, the jump host records the actions or logs of the application. As jump-hosts are non-transparent solutions, they make integration into an existing infrastructure more difficult. Also, the users must use the applications provided by the jump hosts, which may or may not have compatibility issues with their server applications. Auditing of graphical protocols (for example, Remote Desktop, VNC, Citrix ICA) is rarely supported, and even if it is, it can become a performance issue. Transferring files between the server and the client can also be problematic, or not supported at all.

2. Network sniffers are based on switch port mirroring: they receive the network traffic going to the user's servers and try to extract useful information from it. These solutions are easy to integrate and are non-invasive by nature. They also have no effect on the way your users do their work.

However, all this also means that they are very limited in monitoring encrypted traffic, for example, SSH or RDP. Being passive solutions also limits the capabilities of these devices, so they cannot authenticate users, control protocol channels, or terminate unwanted connections to a server.

3. Agent based solutions install agents on the monitored servers that collect information about the users' activities. They are widespread because of the detailed monitoring capabilities they can provide. However, they have some general disadvantages:

• Agents must be installed and maintained on each server.
• Monitoring is limited to the platforms supported by the agent. Typically, they run only on the most common operating systems, leaving other systems and devices (for example, network devices) unmonitored.
• Agents record only access to selected applications, and it is usually not possible to monitor the complete session of the user
• They do not have any control over the connection used to access the server, thus cannot limit their use (for example, they cannot restrict file transfers or port-forwarding in SSH, or file redirection on Windows).

• There is no separation between the monitoring system and the monitored system, and the agents can be manipulated by the monitored superusers. This is essentially the same problem as using the system logs of the monitored system to check the actions of the superuser, who can influence the system logs.

4. Proxy based technologies operate as network proxies or gateways: they are placed between the client and the server, and inspect the traffic on the application level. Since these proxies have full access to the inspected traffic, they have full control over protocol features. For example, you can selectively permit or deny access to certain protocol-specific channels: you can enable terminal sessions in SSH, but disable port-forwarding and SCP, or enable desktop access for the Remote Desktop Protocol, but disable file and printer sharing.

Proxy gateways can operate transparently in the network and are independent from the client and the monitored server. This prevents anyone from modifying the extracted audit information, as the administrators of the server have no access to the proxy gateway.

Certain solutions can even store the audit trails in a time-stamped, encrypted, and digitally signed format, so not even the administrator of the gateway can tamper the audit trails. As transparent solutions, proxy gateways require minimum change to existing IT environment. Also, since they operate on the network level, the users can keep using the client applications they are familiar with and do not have to change their working habits.

Standing in the middle of the monitored network traffic allows proxy gateways to actually intervene in the traffic, making it possible, for example, to require the user to authenticate on the gateway, or to pause the connection until it is authorized by someone appropriate.

With an appropriate way to stream the traffic to the authorizer, the work of the user can be monitored in real-time. It is also possible to extract the files transferred to the server, and store them with the audit trails for later review.

The monitoring and replaying capabilities of PAM solutions show a wide spectrum. Some collect syslog-like log messages, which can be displayed or replayed based on the timestamps of the log messages. Others log only keystrokes. There are solutions that save screenshots from user sessions, or even record the entire session into an AVI file.

THE MONITORING AND AUDITING OF USER SESSIONS SHOULD MAKE IT POSSIBLE TO CONDUCT AD-HOC FORENSICS INVESTIGATIONS, ANALYZE RECORDED DATA IN DETAIL, AND ALSO TO CREATE CUSTOM REPORTS

However, unless some way is provided to process and analyze the content of the screenshots and video files, these are not as useful as they might seem at first.

Movie-like session recording and playback can be a powerful tool, especially for monitoring graphical access (Remote Desktop, XenDesktop, VMWare View), giving auditors the possibility to review all actions of the administrators exactly as they appeared on their monitor.

This can be immensely useful in forensic situations and reporting, if it can be processed automatically to extract the executed commands, applications, the contents of the screen, and other similar information.
To make this happen, advanced PAM solutions index the commands of terminal screens (like SSH or Telnet), and use Optical Character Recognition (OCR) techniques on graphical screens (like in the case of Remote Desktop, XenDesktop, and so on).

The monitoring and auditing of user sessions should make it possible to conduct ad-hoc forensics investigations, analyze recorded data in detail, and also to create custom reports. The subject of the analysis can be a login, a file access, a file transfer, the launch of a program, the stopping of a service and so on.

Access monitoring vs. threat models

Having a PAM-tool in place as described in the previous section greatly increases your ability to monitor working sessions of privileged users or even prevent attacks or any misuse.

Even though the same threat models still exist, discovering exploits and tracking the attacker down becomes much easier. To put it another way: attackers must use much more sophisticated attack vectors to outwit the security measures.

There is not much space to fly under the radar if every user input and screen output is recorded. In addition, privileged access monitoring tools usually authenticate the users to link them to their own account even when they are using shared accounts on the target system, thus making it comparatively easy to respond to the question "Who did exactly what on the system?" Naturally, every access monitoring solution has its limitations or blind spots. Agent-based solutions are usually not capable of recording data transfers from and to the system, or directly monitor activities on non-server platforms.

Proxy gateways and jump-host solutions have no knowledge of anything when the local console is used to access the system, or if the user remotely accesses the system, yet somehow bypasses the monitoring tool.

In order to close the gaps, additional security measures are required, such as well configured firewalls (so that the users can access the servers only through the monitoring tool), effective monitoring of physical system access, and so on. Hence, privileged access monitoring solution can be very powerful as part of a comprehensive security concept.

It's still possible to masquerade malicious activities as harmless, thus making it hard for the investigator to identify certain attacks. Referring to the previously mentioned example of hiding the transfer of the credit card database by using an unsuspicious name like log_messages_for_debugging.tar.gz, proxy-gateway solutions can record the transferred file and make its content available for analyzing possible data breaches.

That is a huge advantage compared to traditional audit logging. Even to use a script to export the database into the file and encrypt it before downloading is not an appropriate way to commit the perfect attack: the script must either be transferred to the target device (in which case the file is recorded), or it must be written on the target device - in a monitored session. A complete recording of all activities makes deception very difficult.

Since users with appropriate rights can erase local logs or stop the local log daemon from sending data to a central log server, using locally running or agent-based access monitoring solutions carries the significant risk of being manipulated or stopped in order to hide certain actions. Although this might raise an alert, there is a huge impact on the traceability of suspected activities in case of an incident.

Cleaning up evidence on a jump-host or proxy-gateway solution is not possible for the attackers, regardless of whether they are insiders or not, as long as the separation of duties principle is part of the company's security policy.

In this case, separation of duties requires that the person performing administration tasks on the target server is never identical with the one responsible for managing the access monitoring solution. And even if the attacker could gain access to the recordings somehow, solutions offering a time-stamped, encrypted, and digitally signed audit trail format would prevent him from accessing and manipulating the recorded data.

Drawbacks of PAM

As with everything else, Privileged Access Monitoring (PAM) solutions have some drawbacks or noteworthy points that you should keep in mind when deploying such a solution. First and foremost, a PAM must be purchased, installed, integrated into your environment, while (you might be tempted to say) logging is already available on your systems, it just needs some tuning and setup.

Depending on your situation and requirements, this might even be true, but do not underestimate the expertise, time, and effort to

properly deploy, setup, and configure a logging system, especially if you are working in a heterogeneous network environment with many different platforms, operating systems, and other devices. Also, PAM systems - especially the ones that record graphical user sessions as well - can cause information overload. It always seems to be faster and easier to do a quick search on some log files than having to find the same information in a bunch of screenshots or video files.

More advanced PAM solutions can cope with this problem by providing powerful search functionalities for the recorded sessions based on session-metadata (for example, username, hostname, date, session length, and so on) and also on session content, listing the executed commands, displayed texts, and other information.

Existing solutions

Below is a collection of free PAM tools, which are useful for web applications and in Linux/UNIX environments, but not for Microsoft Windows. As far as I know, only commercial tools are available for Windows. Also, note that although all of the listed tools are useful, and it is certainly possible to build an activity monitoring system with them, it requires huge effort.

auditd (tinyurl.com/9rs62zb) collects information about the commands executions, as well as various other data, for example, system calls. As a result, auditd generates a huge amount of logs, and you can get much useful information from auditd logs - at the cost of having to do lots of analysis and aggregation to get the desired result. Also, its main aim is to monitor the system as a whole, and not individual user sessions. Auditd is available for most Linux and UNIX systems.

inotifywait (linux.die.net/man/1/inotifywait) is a tool to monitor file system changes. It can be configured to log all kinds of file changes, and can be used especially in forensic situations.

screen (linux.die.net/man/1/screen) is a tool that can be used for the real-time monitoring of terminal sessions. Originally, it was designed to share sessions with other users. It can be started in read-only mode as well as acting as a live monitoring tool. It can be started automatically in certain cases, but the monitored user can easily disable it, so it is more suitable for sharing than reliable auditing.

script (unixhelp.ed.ac.uk/CGI/man-cgi?script) logs the entire screen activity of a terminal. For remote sessions, you can configure it to start immediately after the SSH connection is built, thus logging the entire session. This can be useful in forensic situations, but is not a real-time monitoring tool. Also, it can be easily turned off. rootsh (linux.die.net/man/1/rootsh) is similar to the script tool, and logs the terminal output.

sudoreplay (tinyurl.com/9jrc72p) is an application that processes the system logs of Linux/UNIX systems and replays the commands issued with sudo. Unfortunately, it can be easily bypassed by disabling the display of typed characters in the terminal (that is, disabling echo on tty).

Open Source Tripwire (tinyurl.com/d4pty) is a useful tool for monitoring changes to specific files, for example, important configuration files.

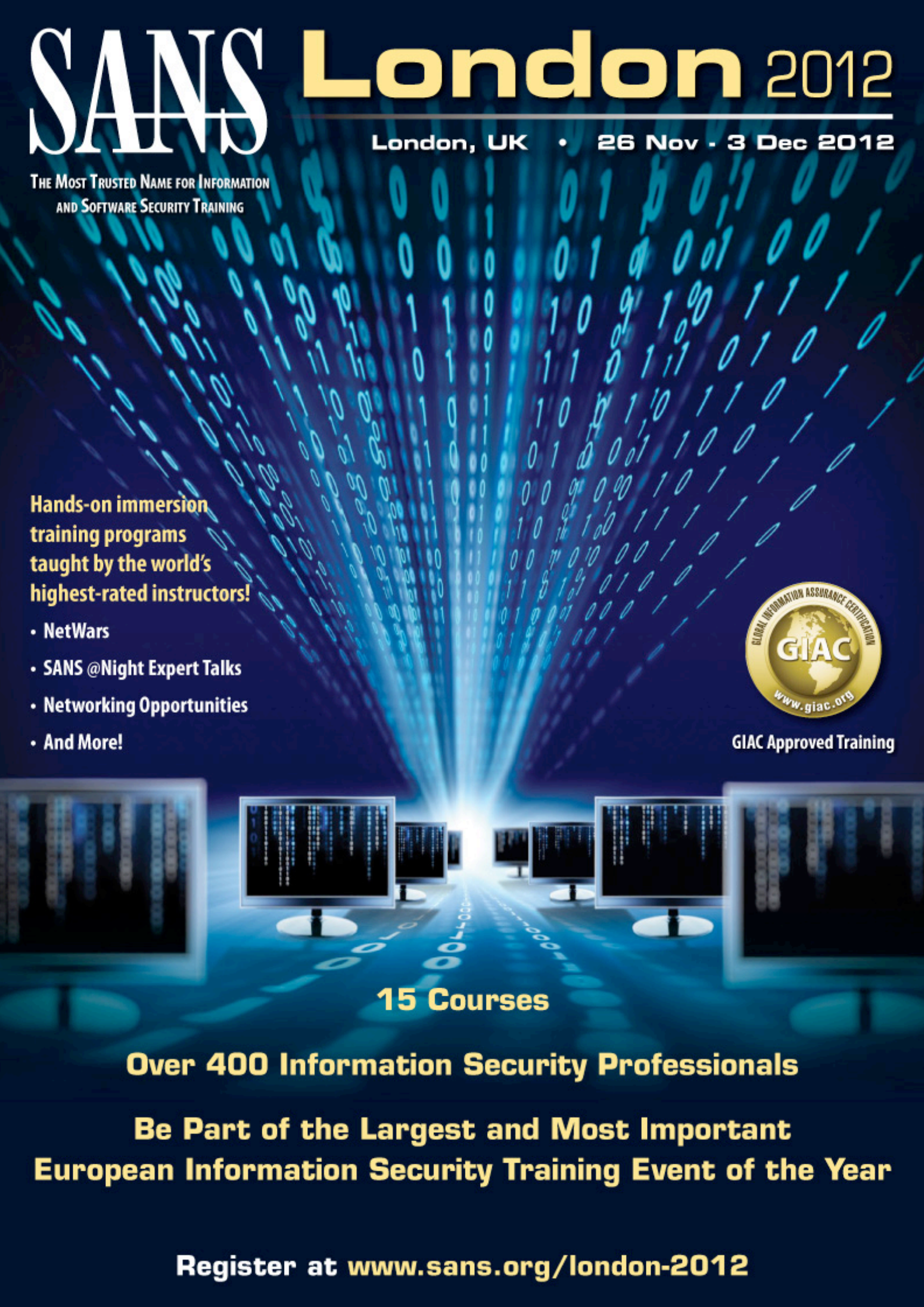
WebAuth (webauth.stanford.edu/) is a free tool to authenticate the users of web applications to an external database - in addition to any authentication required by the web application. This offers a simple way to authenticate the users with their own account, and still allow them to access shared administrator accounts.

Péter Gyöngyösi is the product manager of syslog-ng and syslog-ng Store Box log management solutions at BalaBit IT Security (www.balabit.com). BalaBit IT Security is the developer of syslog-ng trusted logging and Shell Control Box privileged access monitoring solutions. BalaBit bloggers (www.blogs.balabit.com) who contributed to this article: Róbert Fekete, László Szabó, Martin Grauel, Viktor Varga, Péter Czanik, Gábor Marosvári, James Luby.

SANS London 2012

London, UK • 26 Nov - 3 Dec 2012

The Most Trusted Name for Information and Software Security Training



Hands-on immersion training programs taught by the world's highest-rated instructors!

- NetWars
- SANS @Night Expert Talks
- Networking Opportunities
- And More!

GIAC Approved Training

15 Courses

Over 400 Information Security Professionals

Be Part of the Largest and Most Important European Information Security Training Event of the Year

Register at www.sans.org/london-2012

SANS Forensics Prague 2012

www.bit.ly/SANSPrague2012

Angelo Hotel, Prague, Czech Republic

7 October-13 October 2012.

RSA Conference Europe 2012

www.rsaconference.com/events/2012/europe

Hilton London Metropole, London, United Kingdom

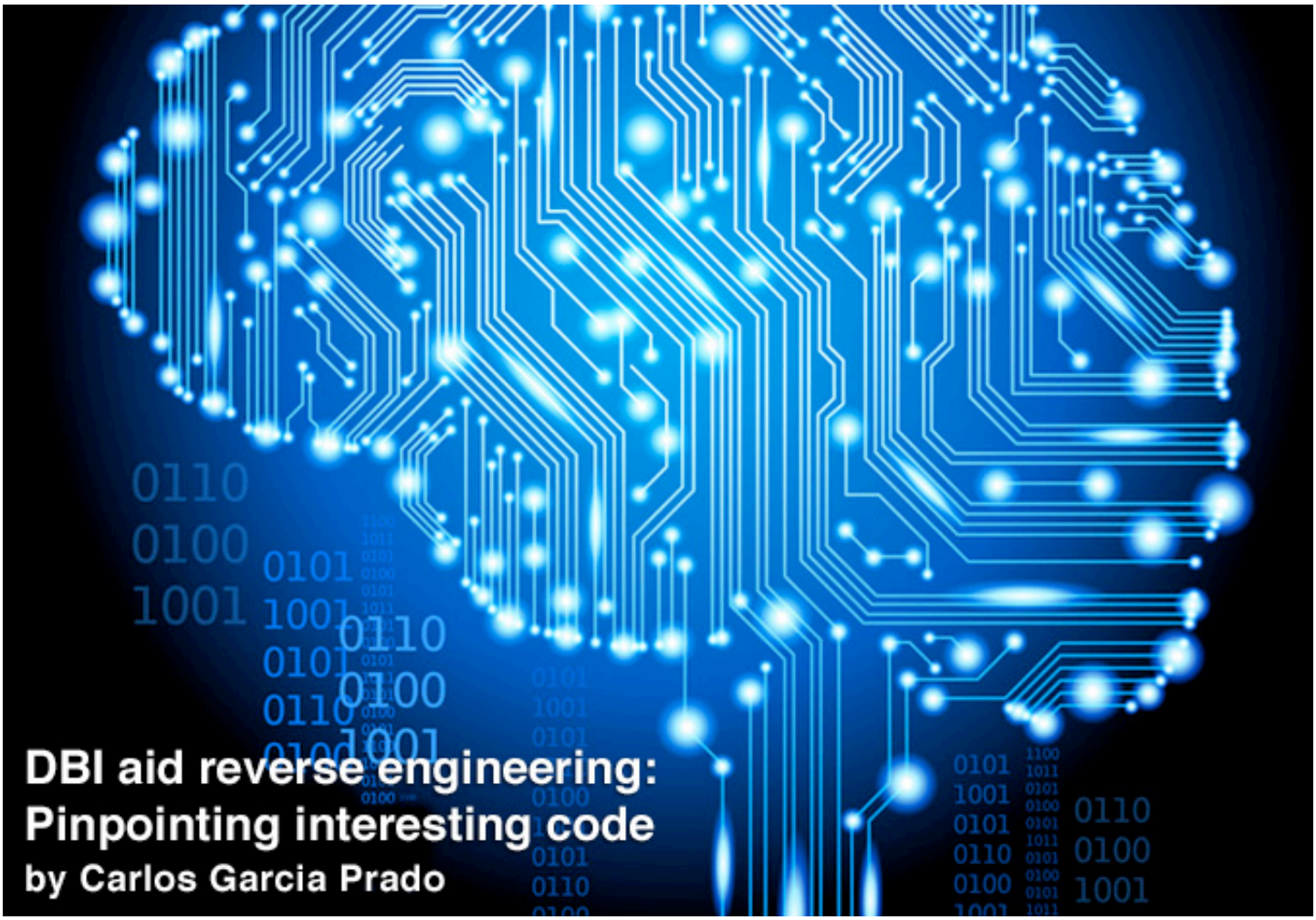
9 October-11 October 2012.

SANS London 2012

www.bit.ly/SANSLondon2012

Grand Connaught Rooms, London, United Kingdom

23 November-3 December 2012.



DBI aid reverse engineering: Pinpointing interesting code

by Carlos Garcia Prado

Compiling is a rather destructive process. Starting with code written in a relatively understandable language for a human being, one ends up with a huge collection of instructions in a format that's easy to digest for a computer processor. Although both are representations of the same problem and will produce the same result, a lot of high-level information is lost in the process.

Binary reverse engineering is the process through which the original information is recovered (to some extent) from this large collection of simple CPU instructions.

Needless to say, this is an extremely complex task and, as researchers, we need as much help as we can get if we want to be able to successfully accomplish it. In this article we will focus on an invaluable tool, namely the Dynamic Binary Instrumentation (DBI).

To be precise, DBI is not actually a tool but a method - an approach for attacking the problem. DBI frameworks expose a rich API that can be used to write our own tools in the classical sense of the word. The purpose of these tools is to analyze and even modify the behavior of a program by injecting small pieces of code into it at runtime.

Through this article I will be using Intel's PIN Framework but there are many others. Some of them have been out there for a long time and are still actively developed. Just look up Valgrind, DynamoRIO or Intel PIN (to name a few) on your favorite search engine.

PIN is my framework of choice

So what is PIN and why am I using it for this article? PIN is Intel's DBI framework. By leveraging its comprehensive API we will be able to inject arbitrary C/C++ code into a running process.

The fact that Intel is behind it reassures me that this project won't (probably) be discontinued in the near future. Along the same lines, there is a lot of documentation and support groups on the web.

I will be focusing on Windows on x86. Being constrained to just one short article there were two possibilities: a broad but shallow - almost theoretical – introduction, or a practical explanation of some nice features along with an example. Also, the rest of the article is structured in a somewhat unusual way. First I am going to explain the practical problem we want to solve and its peculiarities.

After that I'm going to use the PIN API to code a very simple program (pintool) suited to our needs. I will explain the technical details regarding PIN "on demand" when needed.

What's your problem?

When looking for ugly security problems there's no better place to go than a wargame. After all, everything in it has been designed with a single purpose in mind: to drive you mad. A nice example is RCE100.exe, which was part of the famous Nuit Du Hack CTF in 2011. As many other complex problems, it looks easy at first glance.

Figure 1. Hi, my name is RCE100.exe and I came to make your life a living hell.

It consists of a simple window containing what appears to be a text field and a couple of buttons. To solve it you need to find and input the correct password.

Well, that doesn't look that bad, does it? You have already confronted a similar problem a couple of times, for sure. You just need to find the code implementing the text field and button objects, and after that identify the callbacks (code triggered) when (let's say) the login button is pressed.

Following the chain of functions down the path you will eventually get to the function(s) that process your input text. After that it is a simple matter of understanding the check function and crafting a string that satisfies this check. In case you have never done anything similar don't worry, this is not the approach we'll take anyway. I was just trying to give a high level description of a classical, manual reverse engineering train of thought. Basically, find an entry point for your data and manually follow it through all the function calls.

Needless to say, this process is incredibly time/brain consuming even in the best scenario. But in this particular case the problem is even more complex since those just appear to be buttons. The wargame authors simulated buttons, probably replacing them with images or some other object.

Moreover the binary has been packed and is infested by junk instructions, a common anti-reversing mechanism. This, of course, makes static analysis impossible.

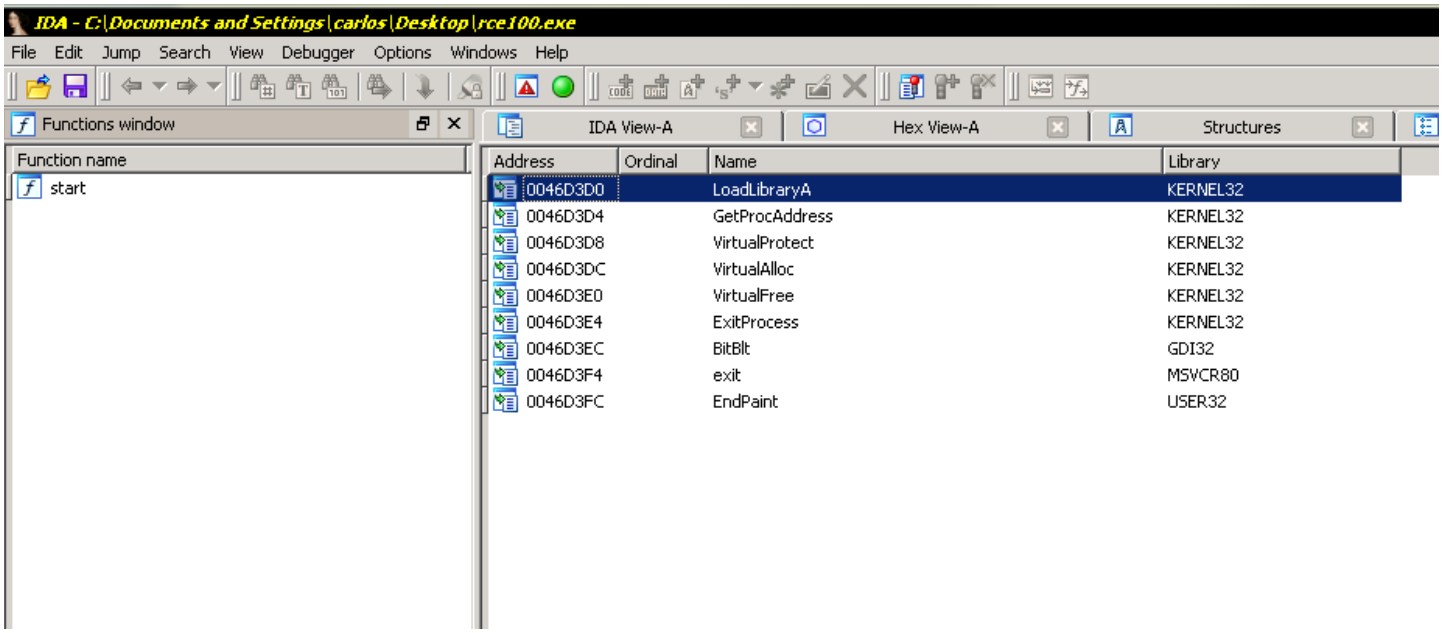


Figure 2. Static analysis shows just one function and merely a handful of imports.

What about dynamic analysis then? Is there anything we can do with a debugger?
It turns out that the usual debugging procedures are going to be difficult in this case, to say the least. It's difficult to find something when you're not sure where to start looking for it.

Since it looks like we are left to redefine our strategy from the ground up, let's try to go for gold here. What if we could pinpoint the exact function(s) responsible of processing our login data by concentrating on reverse-engineering the login check and deducing the condition necessary to pass it?

A way to locate the function(s) implementing a specific functionality is through so-called differential debugging. The idea behind it is very simple: first, find a way to log all the functions being hit during execution to a log file.

Second, run your software and exercise as many code as possible, i.e. click and type everywhere. Use as many parts of the application as you can but leave aside the functionality you are interested in (in this case, logging in).

Last but not least, run your software again, but this time execute only the interesting code. Surely a keen reader is able to see where this is going already. In order to find the interesting functions we are going to use the list on the first log file to filter the superfluous ones out of the second one.

The result is a list of functions that were executed only on the second run and therefore implement the functionality we are interested in. There are some corner cases where this isn't true but we won't discuss them now.

There are already good examples of commercial software capable of doing differential debugging - Zynamics BinNavi to name just one. However, the purpose of this article is to show DBI in action and this is exactly what we are going to do.

The code

Our goal is to instrument the program in such a way that it executes an additional routine every time a function is called. This routine will simply write the address of the function being called to a log file.

Due to length constraints only the relevant snippets will be shown and explained here. The complete code is available for download on my GitHub repository (github.com/carlosgprado/PinTools).

It's worth to note that the following code snippets belong to little more than one of the basic examples distributed with the PIN source code. The real power of the PIN Framework is waiting for you to unleash it.

Let's start with the main function:

```
/* Main function - initialize and set instrumentation callbacks */
int main(int argc, char *argv[])
{
 /* Initialize Pin with symbol capabilities */
 PIN_InitSymbols();
 if(PIN_Init(argc, argv)) return Usage();

 LogFile = fopen("functions_log.txt", "w");

 /* Set callbacks */
 TRACE_AddInstrumentFunction(Trace, 0);
 PIN_AddFiniFunction(Fini, 0);

 /* It never returns, sad :) */
 PIN_StartProgram();

 return 0;
}
```

As you can see, this is rather simple. We begin with initializing PIN. After opening our log file, we set the *callbacks*, i.e. the code implementing the instrumentation itself. Think about these *callbacks* as of hooks. In our case, we want to trace through the code. This will be implemented in the *Trace* function.

Through *PIN_AddFiniFunction* it's possible to specify a function that will be called at the end of the instrumentation process. This usually implements cleanup routines and/or similar ones. The snippet below is a good example.

```
void Fini(INT32 code, void *v){
 fprintf(LogFile, "# EOF\n");
 fclose(LogFile);
}
```

Finally we start the program to be instrumented with *PIN_StartProgram*. From this short discussion it's clear that all the heavy lifting is done by the Trace function so let's check it out.

```
void Trace(TRACE trace, void *v)
{
 /* Do I want to log function arguments as well? */
 const BOOL log_args = KnobLogArgs.Value();

 /* Iterate through basic blocks */
 for(BBL bbl = TRACE_BblHead(trace); BBL_Valid(bbl); bbl = BBL_Next(bbl))
 {
  /* Since a BB is “single entry, single exit” a possible call can only be at
the end */
  INS tail = BBL_InsTail(bbl);

  if(INS_IsCall(tail)) {
   if(INS_IsDirectBranchOrCall(tail)) {
    /* For direct branches or calls, returns the target address */
[a]  const ADDRINT target = INS_DirectBranchOrCallTargetAddress(tail);

    if(log_args) { // Log function arguments as well
[b]  INS_InsertPredicatedCall(tail, IPOINT_BEFORE, AFUNPTR(LogCallAndArgs),
IARG_ADDRINT,
```

```
            target, IARG_FUNCARG_ENTRYPOINT_VALUE, 0, IARG_FUNCARG_ENTRY-
POINT_VALUE, 1,
            IARG_FUNCARG_ENTRYPOINT_VALUE, 2, IARG_END);
    }
    else {
[c]  INS_InsertPredicatedCall(tail, IPOINT_BEFORE, AFUNPTR(LogCall), IARG_AD-
DRINT,
            target, IARG_END);
    }
[…]
```

The trace inspects the program as it's being executed with basic block granularity and is able to recognize events, like for example a function call. When the execution transfers to another function, the target address will be recorded (marked as [a] in the code). This value will be used as an argument in the corresponding callback, i.e. our analysis code. This can be any C/C++ code of our choice but complex code will have a negative effect on the performance, so this has to be carefully considered.

The code snippet showed two callbacks (marked as [b] and [c]) - the only difference between the two being whether the function arguments are logged as well or not. Let's see how one of them is implemented (the other one is almost identical).

```
void LogCallAndArgs(ADDRINT ip, ADDRINT arg0, ADDRINT arg1, ADDRINT arg2) {
 /* If system libraries or already been logged, do nothing */
 if (ip >= MAX_USER_MEM || alreadyLoggedAddresses(ip))
  return;

 UINT32 *CallArg = (UINT32 *)ip;
 /* NOTE: $ has no meaning, just a random token */
 fprintf(LogFile, "$ %p: %u %u %u\n", CallArg, arg0, arg1, arg2);
}
```

This is easy! We check if the address has already been logged or it's higher than an arbitrary value as to avoid logging system DLLs. If the test is negative we are indeed interested in this call and we log this data to a file. Some analogous code regarding indirect calls and other cases has been omitted for the sake of brevity. However, the idea remains the same.

Just for the sake of completion, the *alreadyLoggedAddresses* function is shown below. It checks if the function has been already logged. If not, it saves the address in a vector and returns false. Using this in our callback functions we are able to write to our log file only once per function called, increasing performance and readability.

```
[…]
vector<ADDRINT> loggedAddresses;

/* Auxiliary function. This is a HIT tracer, I want to log every function just
ONCE */
BOOL alreadyLoggedAddresses(ADDRINT ip) {
 if(find(loggedAddresses.begin(), loggedAddresses.end(), ip) !=
loggedAddresses.end()) {
  return true; // item IS IN vector.
 } else {
  /* item is NOT in vector. Push it for the next time. */
  loggedAddresses.push_back(ip);
  return false;
 }
}
```

Once our pintool is ready, it has to be compiled into a DLL. For Visual Studio users, a description of how to set a project can be found at tinyurl.com/6l8935q.
The DLL is consumed by the Pin engine itself. The generic command line usage looks as follows:
C:\Pin_installation\pin.bat –t "C:\path\to \our_pintool.dll" -- "C:\path\to \program_to_instrument.exe"

After running this twice - once without triggering the interesting code and once exercising precisely this functionality - we will end up with two files. The difference between these files will contain the specific functions implementing the functionality we are interested in.

Calculating the difference between the files is as simple as writing ten lines of python but in this case a small tool I wrote can be useful, since it performs - among other functions - exactly this. This python based tool can be found in my GitHub (tinyurl.com/8kozmub).

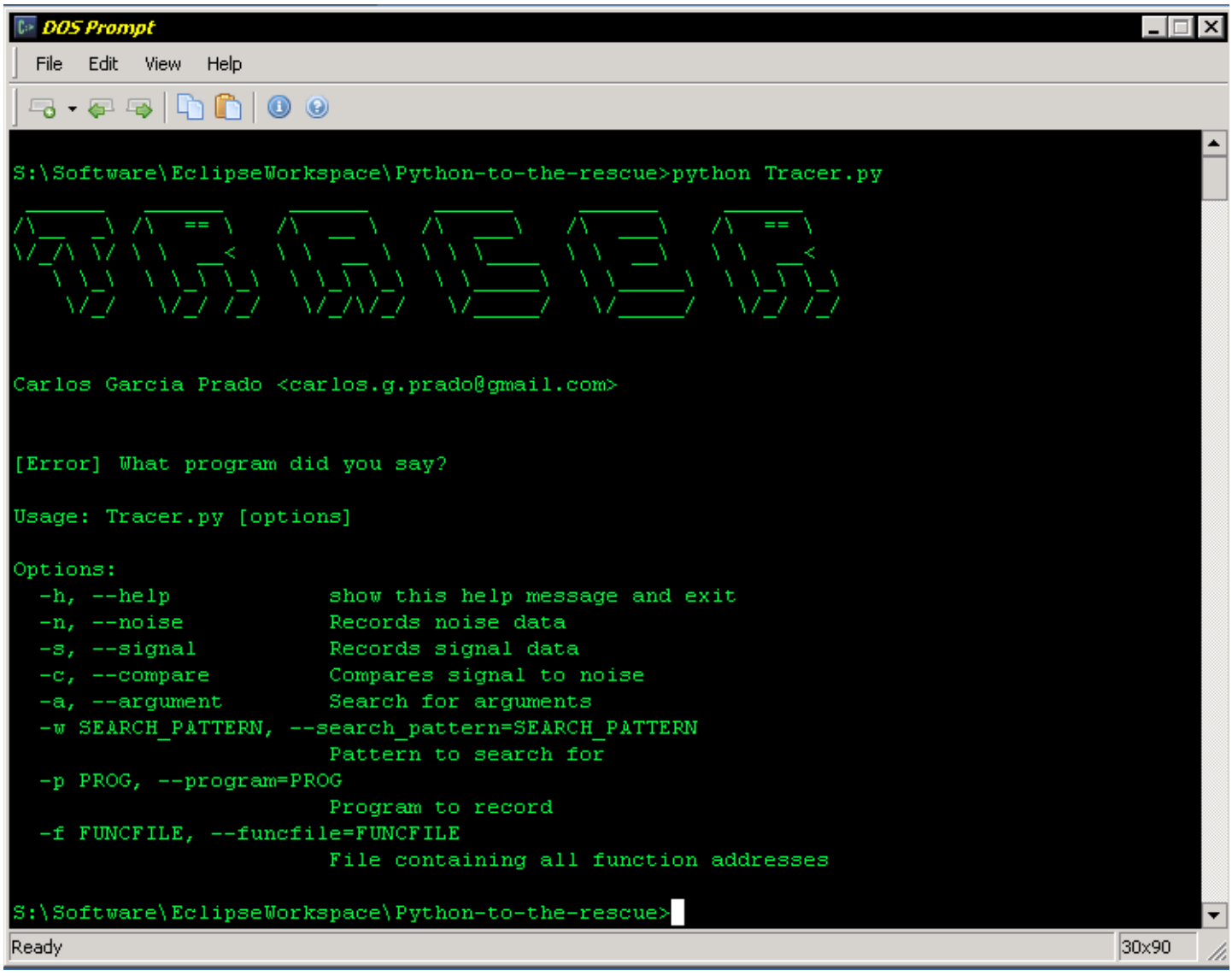


Figure 3. This tool calculates the difference between the output files.

As shown in the figure above, there were only three functions related to the processing of our login string. We have just reduced the original problem to the analysis of three functions.

Let's check if our results are correct. After inspecting the binary inside a debugger we can conclude that the following function does indeed implement the login check.

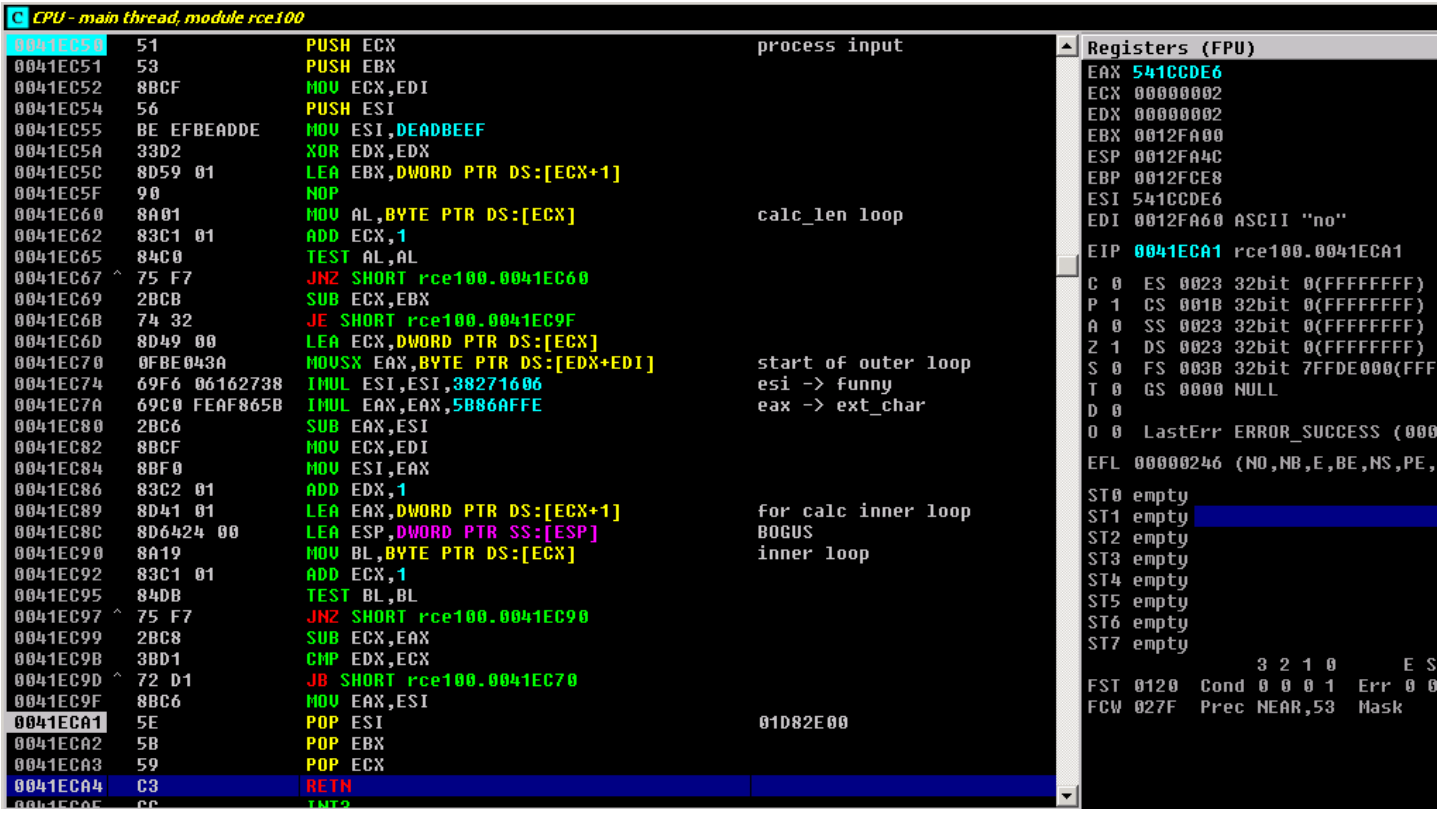


Figure 4. This function process our login and checks its validity.

Note: A keen reader could argue that this function address isn't listed in the results of figure 3. This is because this specific function gets called often in the program but in another context, not related to the login processing. Therefore it's filtered out by the differential debugging process. However, we find it in our analysis when it's called by 0x0041ECB0 with our login string among its parameters.

Rounding it up

Although the rest of the analysis is not strictly related to Intel Pin itself, let's briefly discuss the solution. After all, we have put in a good amount of work to get here, so let's round things up.

The code identified is responsible for checking the validity of our login. It does this by generating a number, based on the numerical value of the login's characters, and comparing this value with a certain hardcoded constant (0xC4B1801C).

The assembly code showed on figure 4 translates roughly to the following C code:

```
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[]) {
 /* RCE100 encoding loop implementation */

 // substr initialized to the whole string
 char *substr = argv[1];
 char *c = substr;
 int ext_char = 0;
 int funny = 0xDEADBEEF;
 int c1 = 0x38271606;
 int c2 = 0x5B86AFFE;
 unsigned int idx = 0;
 unsigned int len_substr = strlen(substr);
```

```
while(idx < len_substr)
{
  ext_char = *c;           // MOVSX
  funny = funny * c1;      // IMUL ESI, ...
  ext_char = ext_char * c2;        // IMUL EAX, ...
  ext_char = ext_char - funny;     // SUB EAX, ESI
  funny = ext_char;        // MOV ESI, EAX
  *c++;                 // MOV ECX, EDI & ADD ECX, 1
  idx++;                // ADD EDX, 1
}
printf("[x] encoded value: %s -> 0x%08x\n", substr, funny);

 return 0;
}
```

This algorithm is difficult (if not impossible) to reverse, so a less refined but effective approach must be taken: brute force.

Since I have a C representation of the algorithm, it can easily be coupled to crunch.c, a very complete string generator (included in BackTrack). After modifying slightly the source code of crunch, compiling and letting it run for a while the solution was found.

Plugging this password into the binary we are granted access and therefore have solved the challenge as shown in the screenshot below.

```
root@yomama[/pentest/passwords/crunch]
[23:43]:./crunch_the_pass 1 10 -f charset.lst lalpha-numeric-symbol14
Crunch will now generate 1094107923781757568 bytes of data
Crunch will now generate 1043422626287 MB of data
Crunch will now generate 1018967408 GB of data
a
aa
aaa
aaaa
aaaaa
aaaaaa
[x] encoded value: gp_gdv -> 0xc4b1801c
^C
```

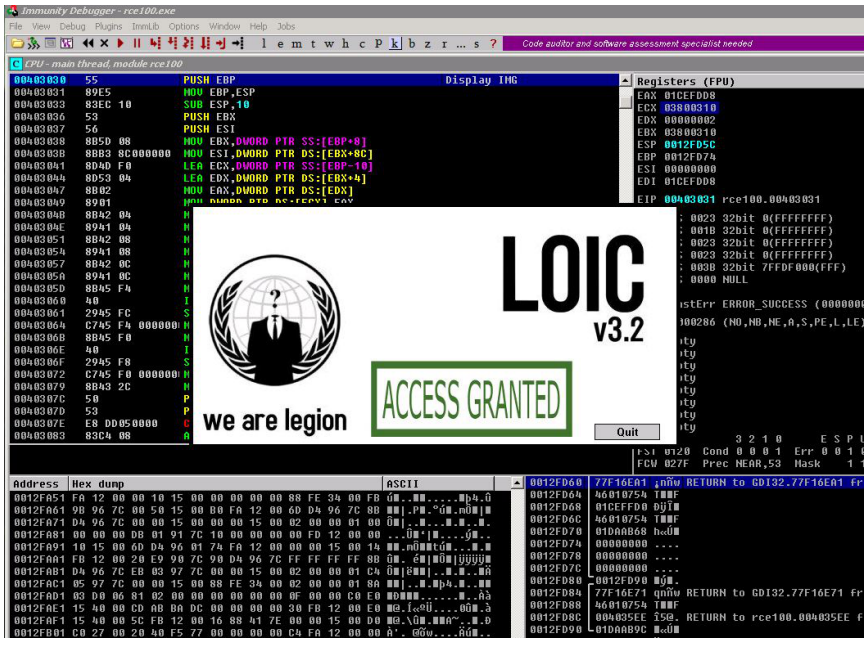


Figure 5. Challenge solved by using dynamic binary instrumentation.

Conclusion

This example has shown only a very limited part of what dynamic binary instrumentation frameworks and specifically Intel Pin are able to do. From here, the limit is basically what you can imagine, from measuring performance to automatically detecting vulnerabilities - and everything in between.

Check out the user's manual online for some cool examples but most important of all, get a copy of Intel Pin and start playing with it.

Carlos Garcia Prado (OSCE, CCNP, other three or four letter achronyms) is an IT security consultant located in Germany. Graduated in Particle Physics, has been involved since eight years in different roles as programmer, teacher, network/firewall professional and pentester.

His actual research interest focuses on making software more secure by identifying vulnerabilities before it hits the market. You can find a comprehensive list of his work and rants at about.me/carlosgprado.

To fully comprehend the importance of data normalization in an Intrusion Prevention System, it is first necessary to understand what data normalization is and what it does, how it accomplishes its goal, and why it is so integral to maintaining security against the advanced evasion techniques used today.

The critical importance of data normalization can also be seen while reviewing security failures and fundamental design flaws in many IPS devices that lack such normalization.

Data normalization explained

Data normalization is the process of intercepting and storing incoming data so it exists in one form only. This eliminates redundant data and protects the data's integrity.

The stored, normalized data is protected while any appearance of the data elsewhere only makes a reference to the data that is being stored and protected in the data normalizer.

The normalizer's job is to patch up the incoming data stream to eliminate the risk of evasion as well as ambiguities. The monitor then views the data in its pure, protected and normalized form. Varying forms of normalization exist on levels of increasing complexity. The complexity is due to the set of requirements that must be met to achieve normalization.

The most basic is known as First Normal Form, which is often abbreviated 1NF. It is followed by Second Normal Form, or 2NF, Third Normal Form, or 3NF and can continue increasing in forms and complexity as required or desired.

Normalization benefits

Normalization plays a key role in the security of a network, provided that normalization extends to every protocol layer. One of the major benefits is the forced integrity of the data as the data normalization process tends to enhance the overall cleanliness and structure of the data. Normalization significantly contributes to the fortification of a network, especially in light of typical networks' three main weak points: traffic handling, inspection and detection.

Where many IPS devices go wrong

When it comes to traffic handling, many IPS devices focus on throughput orientation for the most rapid and optimal inline performance. This process, while attractive for its rapidity, makes it impossible for full normalization to take place. The data traffic is then inspected without normalization, offering prime opportunities for infiltration to take place. One may agree that a rapid and optimal output performance is useless if the payload is riddled with malicious invaders.

When many IPS devices do employ normalization, they often rely on shortcuts that only implement partial normalization as well as partial inspection. This leaves gaps in the security and provides optimal opportunity for evasions.

TCP segmentation handling is one example of such a process, as it is only executed in chosen protocols or ports and is drastically limited in its execution.

Shortcut exploitation is a familiar evasion method and, with the proliferation of IPS devices that fail to perform full normalization, it is likely to remain that way due to its ease of execution.

Normalization significantly contributes to the fortification of a network, especially in light of typical networks' three main weak points: traffic handling, inspection and detection.

Many IPS devices fall short in other areas, as well. They often perform only a single layer of analysis, execute traffic modifications and interpretations and rely on inspection of individual segments or pseudo-packets. Their detection methods are based on vulnerability and exploits, banner matching or shell detection.

Their updates are generally delayed and their evasion coverage is extremely limited. Evasions can easily exploit the limited inspection scope by spreading attacks over segments or pseudo-packet boundaries.

Packet-oriented pattern matching is insufficient as a means of invasion detection due to the need for a 100% pattern match for blocking or detection. Advanced Evasion Techniques (AETs) possess the ability to utilize a vast multitude of combinations to infiltrate a system, rendering the likelihood of a 100% pattern match for every possible combination nonexistent. It is simply impossible to create enough signatures to be effective.

AETs exploit the weaknesses in the system, often being delivered in a highly liberal manner that a conservatively designed security device is incapable of detecting.

In addition to using unusual combinations, AETs also focus on rarely used protocol properties or even create network traffic that disregards strict protocol specifications.

A large number of standard IPS devices fail to detect and block AETs, which have therefore effectively disguised a cyber attack that infiltrates or even decimates the network. Standard methods used to detect and block attacks generally rely on protocol anomalies or violations, which is no longer adequate to match the rapidly changing and adaptable AETs.

In fact, the greatest number of anomalies occurs not from attacks, but rather from flawed implementation in regularly used Internet applications.

An additional issue that arises with many IPS devices is the environment in which they are optimized. Optimization typically takes place in a clean or simulated network that has never suffered a complex and highly elusive attack.

Resistance to normalization

Resistance to data normalization does not typically arise from the advanced security it promises, but rather the impact it may have on a network. When the security design flaw is found in hardware-based products, network administrators may resist the upgraded security measure due to the necessity of significant research and development for redesign. Additional memory and CPU capacity are also required to properly implement a data stream inspection that comprehensively protects against AETs.

When vendors decide that the required changes are impossible to implement, they leave their networks highly vulnerable to exploits and attacks.

Focusing on the cost of the cleanup required for all infected computers in the network, and the even higher cost of network downtime, can help change the minds of vendors who continue to resist the necessary adaptations.

How the most effective IPS devices use data normalization

Instead of analyzing data as single or combined packets, effective IPS devices analyze data as a normalized stream. Once normalized, the data is sent through multiple parallel and sequential machines. All data traffic should be systematically analyzed by default, regardless of its origins or destination.

The most effective way to detect infiltration is to systematically analyze and decode the data, layer by layer. Normalization must occur at every layer simply because attacks can be hidden at many different layers.
In the lower protocol layers, the data stream must be reconstructed in a unique manner. Modifications should generally be very slight or nonexistent, although any fragments or segments containing conflicting and overlapping data should be dropped.

Normalizing traffic in this manner ensures there is a unique way to interpret network traffic passing through the IPS. The data stream is then reassembled for inspection in the upper layers. Inspection of constant data stream in this manner is a must for correcting the flaws and vulnerabilities left open by many IPS devices. This process also removes the possibility of evasion of attacks that span over segment boundaries.

Higher levels are subjected to inspection of separate data streams that are normalized based on the protocol. In compressed HTTP, for instance, the data can be decompressed for inspection.

In another example, MSRPC-named pipes using the same SMB connection would be demultiplexed and inspected separately.

Such a thorough and comprehensive data normalization process is the most effective way to protect networks from AETs and other threats that may otherwise disguise themselves to go undetected through standard IPS.

The most effective IPS devices will ensure evasions are removed through the normalization process before the data stream is even inspected.

This normalization is so successful because it combines a data stream based approach, layered protocol analysis and protocol specific normalization at different levels. It therefore helps fortify a network's three weakest points and keeps malicious invader’s attacks at bay.

Darren Suprina is an IT systems designer and security professional with more than 30 years of experience working at Stonesoft (www.stonesoft.com). This has included intellectual property creation, research, development, software and infrastructure validation, systems auditing, work as a professional witness, and author. He has previously provided line of business creation, team leadership, and engineering value while at Agilysys, AuthentiDate, Sun Microsystems, SIAC, and Intel.

HELP NET SECURITY

www.net-security.org

14 years of information security news