Emerging Security Threats

Mark Loveless

April 20, 2010

About Me

- I do not work for a security boutique or consultancy, software or hardware vendor
- I have not written a book I am trying to sell
- I am not trying to drum up business nor recommend business to a friend or a friend's company

Agenda

- Definitions of 0day, botnets, “decent” spear phishing, client-side attacks, and APT
- Attacks and why you should not worry about APT any more than any other attack scenario (which means still worry a lot)
- Mitigation techniques

- This will get technical, sorry, it is a complex world now

Definitions

Basics

- Attacker – the bad guy, the adversary; the attacker is the person attacking you, plain and simple
- Vulnerability – a flaw that exists in software, hardware, or even communication protocols that has security implications
- Exploit – a series of steps, techniques, or even a software program designed to take advantage of the vulnerability

Basics (cont)

- Clicker – mouse-happy victim of email or URL-based attacks
  - Example: "Logs show we received 13 emails with the bad link." *"How many clickers?"* "3 people apparently clicked on the bad link."
- Command and Control (C2) – common designation for communications between the attacker and a compromised computer system or systems that pass instructions and information
- Exfiltration – the act of pulling data from the compromised system or systems back to the attacker

Quick Definition - 0day

- Traditional definition in use here
- The vulnerability has been public for exactly zero days
- Once the vulnerability has been exploited *and figured out* it is no longer 0day
- Once the vendor issues a patch, it is no longer 0day, as attackers can reverse engineer patches

Weaponized vs Proof of Concept

- **Proof of concept** implies the code will potentially crash the target (this includes a lot of MetaSploit Framework modules), especially due to minor differences (languages, test environment, etc)
- **Weaponized** means the exploit will not cause the target application or system to crash, memory corruption is cleaned up, etc so the victim is not alerted
- Development time on PoC can be hours, full weaponization can take days or weeks

The Botnet

- Quick definition – a series of computers linked together under a single adversary's control who can direct these computers to perform tasks
- Historically used for DDoS and spam, were often controlled primitively via IRC etc
- Modern botnets can operate independent of a single operator, use encryption, P2P-like C2 protocols, can probe and attack, include update mechanisms

Phishing

- Phishing is a sloppy technique used to entice victims into clicking and giving up credentials etc. typically via an email
- Decent phishing is where the phishing email is sent within proper context, no misspellings, and is forged rather well
  - Targeted phishing is known as spear phishing, and usually involves text that is specific to the target (e.g. military jargon email with military-related doc sent to DoD contractor, often apropos to the contractor's current project)

The Attack

Attack Techniques

- Client-side attacks are number one
- The number one targeted OS is Microsoft Windows
- Easiest target? Send in an executable as an attachment
- Second easiest? Send in an attachment that when opened compromises a common application
- Third? Link to a website which does essentially the first two or exploits the browser itself

When 0Day Is Used

- The adversary wants inside due to a perceived need that the target (or the target data) and its value is tied to time
- Adversary intelligence suggests certain social engineering scenarios will be more effective at certain times
- The vendor has issued a patch and the 0day is no longer 0day, and is now limited in value

The 0Day Pipeline

- Adversaries will typically have more than one 0day in development, and at least one fully weaponized
- Secondary exploits will be non-0day exploits for vulnerabilities without known public exploit code (typically no IDS/IPS signature, or signature limited in scope)

What Else Is Used

- Heavily-obfuscated JavaScript
- C2 channels are encrypted
- Exfiltration of data uses varying levels of stealth
- Client-app specific obfuscation (think PDF evasion techniques in MetaSploit Framework)

What is APT?

- Advanced Persistent Threat
  - Coined by the U.S. Air Force a number of years ago, term has been in use for a while in .mil/.gov circles
- Advanced mean they are not script kiddies
  - Adversary is more thoughtful, thinks (plans) before acting
  - Does not mean 100% effective or are world's best
- Persistent does not mean blindly relentless
  - The target is well probed, well considered, and as the adversary gains new (or the latest) tricks, fresh attempts are made

Typical APT Scenario

- Adversary has custom RAT (Remote Administrative Tool) sitting on a server
- Adversary develops client-side attack software
- Payload of exploit grabs the RAT and runs it
- RAT initiates connection back to the adversary, but otherwise functions as a server for the adversary's Command and Control (C2)

Other APT Characteristics

- Ordinary attacker may stop after being detected
- APT attackers will use intelligence gathered for the next attack
  - E.g. New attack, RAT has your proxy server IP address hard-coded in, error messages include internal server names
- APT attackers will hide in noise, let you find "easy" compromises so you think you have them all
  - Usually in .mil/.gov/contractor scenarios

APT - What Else?

- The adversary is patient
- The adversary can hide themselves from the obvious detection methods in a lot of cases
- The adversary operates effectively, assumes they will get caught eventually, and acts accordingly

"Old School" APT

- Remote access to a system, initiated by the attacker against a server, client-side attack, or client system exposed to the Internet
  - Or dialup, VPN, etc
- Once in, install backdoors and clean logs (the old school definition of "rootkit")
- Harvest system for additional targets within the organization

Modern APT

- Same as old school APT, except the main point of entry is client-side attacks
- There is also a specific objective associated with the target

Why This Is New

- Many elements have been brought together
  - An attack "component" often mirrors sophisticated techniques
  - Multiple sophisticated components will make up one attack scenario
- Social engineering
  - Advances in social networking coupled with creative Googling help provide frameworks for who knows who and who works together
  - Spear phishing is done via well crafted emails with knowledge of who the victim might know (and trust), contextually correct scenarios, and properly spell-checked

Why This Is Really Not New

- All techniques used have been discussed at security and hacker conferences for years
- Privacy advocates have warned about the dangers of providing too much information in social networks, which could lead to better socially engineered attacks
- Although some of the initial penetration techniques seem novel, the intrusion follows the usual pen test blueprint (as illustrated in numerous books and white-hat hacker training classes)

What We Know About APT

- The “infrastructure” (RAT on server, JavaScript “front end” to exploit, etc) is already developed and in place, sometimes weeks before the attack
- Different components of the infrastructure use different coding styles and levels of sophistication, implying different authors, perhaps specialized it certain techniques
- Not all exploits are 0day
- Exploits are typically fully weaponized

Perspective

- Remember the high profile attack against Microsoft where their network was compromised? In 2000? Early example of what we would now call APT.
- A lot of hype over APT, but in certain circles it is used as shorthand
  - "We were compromised." *"How bad?"* "4 clickers, malware installed." *"Damn."* "Well at least it wasn't APT, just run-of-the-mill malware."

Mitigation

Common Defenses Thought To Work

- Anti-Virus
- IDS/IPS
- Firewall

Anti-Virus Good Points

- Swats down the easy flies
- Shows an auditor you can meet minimal requirements towards due diligence
- Requires attackers to use obfuscation techniques (more on that later)

IDS/IPS Good Points

- Swats down the easy flies
- Shows an auditor you can meet minimal requirements towards due diligence
- Requires attackers to use obfuscation techniques (more on that later)

Firewall Good Points

- Swats down the easy flies
- Shows an auditor you can meet minimal requirements towards due diligence
- Requires attackers to use obfuscation techniques (more on that later)

Why These Technologies are Ineffective

- Attackers will use techniques like multi-pass packing, encryption, and other forms of obfuscation to bypass A/V and IDS/IPS
  - Not just the major brands, ALL of them (they test this out first)
  - Host-based IDS is marginally better, but not much
- Your firewall allows bad stuff in, via email, web browsing, etc
  - It is cheaper to use router ACLs on your border routers than fancy firewalls, unless they have other features you need
  - However you may not pass an audit

New Ideas

- Indicator Classification
- Intrusion Chain
- “Reactive” vs “Proactive”

Indicator Classification

- Atomic indicator - Indicator which may or may not be representative of exclusive adversary activity.
  - Includes IP addresses, email addresses, host names, domain names, strings of text used in C2 or email subject lines.
  - Possible these indicators have non-adversary activity associated with them, prone to false positives.
  - Determination of the validity of these indicators will often require additional analysis.
- Computed indicator - Indicator which is based upon static data and is computed from that data, usually hashes, dropper file names and locations of malware files.

Indicator Classification (cont)

- Behavioral indicator – A sum total of multiple indicators that tell a story
  - Example: "Adversary sends HTML mail with subject line 'New DoD paper' with link to blah.blah/blah.htm and uses CVE-2010-XXXX to drop an EXE (hash blah) in directory c:\blah, phones home to 1.2.3.4 over UDP port 53 with string “blah” in packet.“

Intrusion Chain (part 1)

- Reconnaissance – Adversary search for info for the attack
  - Probing to determine if IPS/IDS is in place
  - Spidering of web servers for "boilerplate" text for use in phishing attacks etc.
- Weaponization - Quality and expertise behind the exploit/malware
  - Coding styles and level(s) of sophistication can determine number of adversaries
- Revealed in post-intrusion analysis.

Intrusion Chain (part 2)

- Delivery - Payload delivery method to the target
  - Intersection of defensive front lines and adversary
- Compromise / Exploit - Vulnerabilities being exploited, software (common), hardware (rare), or human (common, e.g. social engineering)
  - Complex in multi-stage payloads, failure of any stage of the overall exploit can result in failure for the adversary
  - Includes lateral compromise within the target environment
- Detection is indicative of an attack attempt, and possible partial success in multi-stage payloads scenarios

Intrusion Chain (part 3)

- C2 - Command-and-Control phase represents the period after which adversaries leverage the exploit of a system
  - Rootkit installed
  - Includes further uploads (malware, hacking tools, etc)
  - Further lateral compromise within the target environment, aka traditional “penetration test” scenario
- Exfiltration - Data copied from victim to adversary
  - Data about network and working environment is exfiltrated
  - May be specific documents involving specific projects
  - Can involve extremely large-scale amounts of data being copied offsite by the adversary (all PDF, DOC, PPT, and XLS files; entire SQL databases, etc)

- Detection here is considered the worst part of the intrusion chain, indicative of full and complete compromise

Reactive vs Proactive Approaches

- "Reactive" – Work from the back of the intrusion chain, post-intrusion searches of indicators
- "Proactive" – Work more from the front of the intrusion chain, analysis of patterns of indicators is the driver
- You will have to do both, the hardest is doing any proactive work

Basic Things I Will Not Cover

- Smart firewalling, antivirus up-to-date, good IT/Security policy in place, user awareness training, regular pen tests
- You should be doing ALL of this already

Intelligent Use of IDS/IPS

- Common IDS/IPS may have 2500+ signatures
  - False positives create more work for analysts
  - Noisy signatures get ignored
- Only use most relevant signatures
  - If you are patched for the XYZ flaw, you don't need the signature
  - Internal attack signatures should not be on the perimeter, and vice versa

Patching

- Patching may break things, but even testing may not discover this
- Get good at handling issues when patching breaks things, and patch immediately without testing
- Adversary will attack with 0day *immediately* if a patch is released for it

More Proactive Steps

- Turn on any firewalling features on the client systems
- Use proxies and web filtering
- Use DNSBL
- Use local DNS blackholing
  - Can be used to detect infected systems, or clickers
- Enable ASLR/DEP
  - Could kill initial exploit, cutting off the chain
  - Most attacks in the wild still do not use ASLR/DEP bypassing (expect this to change)
- Disable scripting languages where possible
  - JavaScript in Adobe Acrobat is a big enabler

LOGS!

- Central logging of processed data
  - SMTP
  - Proxies
  - Key firewall denials (not all)
  - Key DNS lookups (not all)
- Logs should be searchable
  - grep is fine, even scriptable
- Try to be as near realtime as possible
- Logging server must be 100% secured, no exceptions or compromises

GATHER INTEL!

- Keep a database of indicators, past intrusion attempts and events
- Use these indicators to sift through logs and develop new indicators
  - E.g. whois analysis
- Age indicators as appropriate
  - If Mom and Pop's Web Site is compromised and the source of an intrusion, you do not need to keep it around in block lists for 9-12 months
- A ticketing system with workflow capabilities is great for not just organizing analysis work, but for helping to gather intel

FLOWS!

- Traffic flow analysis is great for finding bad stuff
  - If Bob in Accounting's computer starts talking heavily to domain controllers and neighbor's computers, Bob may be owned
  - Exfiltration of data in large quantities stands out like a sore thumb
- Don't use outside the perimeter unless you are small or very very very bored
  - Inside is best, easier to catch things

If You Can Afford It…

- A working sandbox to test malware
  - Reverse engineering is even better
- Separate Internet connection just for analysis work
  - Don’t tip off the bad guys

Use Your Tradecraft

- Analysts should encrypt communications between each other
  - Assume adversaries will read your analysts' email, because the better ones will
  - Use encrypted chat on locked-down servers, e.g. SILC

Invest In People

- Don’t spend six figures on each piece of every fancy software with three overworked analysts chasing false positives
- Spend those six figures on very smart people
  - IDS/IPS specialist who can write their own custom signatures using freeware IDS/IPS software and tools
  - Traffic flow expert who actually understands network protocols and can *use* that expensive software
- Let them play, and pay them to go learn at security/hacker conferences
  - ALL the “latest” techniques and tricks used in modern attacks were first presented at hacker conferences. ALL of them.

Questions?

- Contact:
  - thegnome@nmrc.org
  - mloveless@mitre.org