Common Threats

COM-402: Information Security and Privacy

(slide credits: Philipp Jovanovic, Linus Gasser)
Outline

- Overview of threats to IT-systems
- Cyber attack lifecycle
  - Commodity threats
  - Hacktivism
  - Advanced Persistent Threat
- Classes of non-physical threats
  - Social engineering
  - Software vulnerabilities
  - Side-channel attacks
  - Distributed Denial of Service
  - Malicious Software (Malware)
Threats

● Definition (ISO27005):

“A potential cause of an incident, that may result in harm of systems and organization.”

● Definition (NIST FIPS 200):

“Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.

Also, the potential for a threat-source to successfully exploit a particular information system vulnerability.”
Overview of Threats to IT-Systems

Physical:
- Environmental
  - Fire, water, pollution
  - Earthquakes
  - Volcanic eruptions
  - Cosmic radiation
  - War, riots
- Loss of essential service
  - Electrical power
  - Air conditioning
  - Telecommunication
- Technical failures

Non-physical:
- Social engineering
- Software vulnerabilities
- Side-channel attacks
- Distributed Denial of Service
- Malicious Software (Malware)

> This lecture focuses on the non-physical security threats
Exploits

● From Wikipedia:

An exploit (from the English verb to exploit, meaning "using something to one’s own advantage") is a piece of software, [...] that takes advantage of a bug or vulnerability [...] gaining control of a computer system, [...] or a denial-of-service (DoS or related DDoS) attack.

● Can be used by malware to gain control of a system

● Exploiting vulnerabilities of
  ○ Humans - social engineering
    ■ Phishing (broad) or spear phishing (targeted): trick user into clicking/activating malware
    ■ Physical-world: e.g., call helpdesk and impersonate internal authority figure
  ○ Software bugs
    ■ Known exploits: cheap/ubiquitous in underground economy, target unpatched devices
    ■ 0-day exploits: expensive ($500K bounty for iOS), often hoarded or used sparingly
  ○ System attacks
    ■ Distributed Denial of Service (DDoS)
    ■ Side-channel attacks
The Value of a Hacked PC

Source: KrebsOnSecurity
Outline

● Overview of threats to IT-systems
● Cyber attack lifecycle
  ○ Commodity threats
  ○ Hacktivism
  ○ Advanced Persistent Threat
● Classes of non-physical threats
  ○ Social engineering
  ○ Software vulnerabilities
  ○ Side-channel attacks
  ○ Distributed Denial of Service
  ○ Malicious Software (Malware)
Cyber Attacks Lifecycle I

Depending on the attack type some or all of the following steps are involved:

1. Preparation
   - Define target, from broad (everyone) to focused (individual)
   - Find and organize accomplices
   - Build and/or acquire tools
   - Research target (infrastructure & people)
   - Prepare “watering holes” (traps)

2. Gain Access
   - Deployment (social engineering, exploits, etc.)
   - Initial intrusion
   - Privilege escalation

3. Maintain Access
   - Strengthen foothold (install rootkits, etc.)
   - Setup stealthy access methods
   - Internal reconnaissance
   - Expand access (lateral movement)

Depending on the attack type some or all of these steps are involved:

4. Complete Mission
   - Exfiltrate data
   - Damage target
   - Use target as starting point for other attacks
     - Spam
     - Distributed Denial of Service
     - Click fraud
     - …

5. Cover Tracks
   - Delete log files
   - Modify logging applications
   - Memory-only persistence

Commodity Threats

- Non-targeted ("shotgun" approach)
- Usually non-stealthy and fully automated
- Goal is often short-term financial gains
- Often considered low risk to attackers
- Increasingly a starting point for more sophisticated attacks

Forms:

- Computer worms
- Malicious ads
- Malware-infected spam
- ...
Example: London Stock Exchange

Sunday, 27 February 2011

London Stock Exchange hit by malware

The London Stock Exchange website exposed some visitors to drive-by malware attacks today. Merely viewing the homepage at www.londonstockexchange.com (without clicking on anything) caused my Windows computer to be compromised by malware. This malware was apparently delivered through third-party advertisements which appeared on the site.

The malware was a classic spoof antivirus program which used a software vulnerability to download and install native executable code. The spoof program appeared in the system tray and prevented other processes such as Task Manager being run, falsely claiming that they were infected with a virus. The malware then tried to extort payment to fix the artificial problem it had created. It also replaced the wallpaper image with the following message:

![Warning Message]

Source: High Severity
Hacktivism

Hacktivism

Controversial term with several meanings such as:

- Politically motivated hacking
- Variant of (anarchic) civil disobedience

Forms:

- Software (e.g., PGP)
- Website mirroring to circumvent censorship
- Website defacement (e.g., various Anonymous / Lulzsec incidents)
- Anonymous blogging
- Distributed Denial of Service
- ...
“Hacktivism” Example – Sony Pictures

Hackers shut down Sony Pictures' computers and are blackmailing the studio

Source: The Verge

- Nov. 2014: “Guardians of Peace” (GOP) leak confidential data of Sony Pictures
- Dec. 2014:
  - GOP demand that Sony cancels release of The Interview, a comedy movie on a plot to assassinate North Korean leader Kim Jong-un
  - FBI attributes attack to North Korea
  - Former employees sue Sony over leak
- Dec. 2015: Sony pays ~$8 Million in settlement of the lawsuit

Estimated costs for Sony: $35+ million
Advanced Persistent Threats

Advanced Persistent Threats (APTs)

- **Advanced:**
  - Targeted multi-step attacks
  - Utilize full spectrum of available intrusion techniques
  - Often use specialized tools
  - Combine multiple attack vectors (90%+ attacks start with spear phishing)

- **Persistent:**
  - “Low-and-slow” approach
  - Prioritize long-term over short-term goals (e.g., immediate financial gain)
  - Continuous monitoring and interaction (avg attack duration: 1 year; known max: 5 years)

- **Threat:**
  - Human-coordinated attack
  - Attackers are skilled, motivated, and well-funded; have a clear goal (e.g., industry espionage)
  - No “fire-and-forget” approach as core component (e.g., fully-automated malware)
Chronology of the attack


- **Dec. 2015**: First hints (IP’s) from external organisation. No in-depth search possible because proxy does not log internal client IP’s.


- **22.1.-31.1.2016**: Incident opened by MELANI/GovCERT.ch and RUAG.

- **1.2.-29.2.2016**: Monitoring established.

- **1.3-30.4.2016**: Enhanced Monitoring established.

- **3.5.2016**: Several press reports about the incident. This leakage heavily damages the ongoing investigation, rendering the ongoing monitoring useless.

Source: MELANI
Clicker-time - #1

What is your take on ‘Hacktivism’?
Outline

● Overview of threats to IT-systems
● Cyber attack lifecycle
  ○ Commodity threats
  ○ Hacktivism
  ○ Advanced Persistent Threat
● Classes of non-physical threats
  ○ Social engineering
  ○ Software vulnerabilities
  ○ Side-channel attacks
  ○ Distributed Denial of Service
  ○ Malicious Software (Malware)
Description of non-physical threats

- **Social engineering**
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)

**Social Engineering**
- Psychological manipulation of people tricking them into taking actions that benefit the attacker (e.g., reveal confidential information)
- Often among the first steps in a cyber attack
- Forms:
  - (Spear) Phishing
  - Coercion
  - Watering holes
  - Baiting
  - ...
Social Engineering

TEEN WHO HACKED CIA DIRECTOR’S EMAIL TELLS HOW HE DID IT

A hacker who claims to have broken into the AOL account of CIA Director John Brennan says he obtained access by posing as a Verizon worker to trick another employee into revealing the spy chief’s personal information. Source: Wired

Ubiquiti Networks Inc., the San Jose based manufacturer of networking high-performance networking technology for service providers and enterprises, announced in its fourth quarter fiscal results that it was the victim of an email business fraud incident resulting in the loss of $39.1 million dollars. Source: CSOonline

Spoofed communications from executives at the victim firm in a bid to initiate unauthorized wire transfers.
Social Engineering

Users Really Do Plug in USB Drives They Find

Matthew Tischler1  Zakir Durumeric1  Sam Foster1  Sunny Duan1
Alec Mort1  Elie Bursztein2  Michael Bailey2

1 University of Illinois, Urbana-Champaign  2 Google, Inc.

Abstract—We investigate the anecdotal belief that end users will plug and play in USB flash drives they find by conducting a controlled experiment in which we drop 297 flash drives on a large university campus. We find that the attack is effective with an estimated success rate of 45–98% and expeditions with the first drive connected in less than six minutes. We analyze the types of drives users connect and survey those users to understand their motivation and security profile. We find that a drive’s appearance does not increase attack success. Instead, users connect the drive with the altruistic intention of finding the owner. These individuals are not technically incompetent, but are rather typical community members who appear to take more recreational risks than their peers. We conclude with lessons learned and discussion on how social engineering attacks—while less technical—continue to be an effective attack vector that our community has yet to successfully address.

Concerns about USB security are real: 48% of people do plug-in USB drives found in parking lots

Source: Elie Burszstein’s blog

Source: BH USA 2016

Attacks pros & cons

<table>
<thead>
<tr>
<th>Attack vector</th>
<th>Mostly used by</th>
<th>Complexity &amp; Cost</th>
<th>Reliability</th>
<th>Stealth</th>
<th>Cross OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social engineering</td>
<td>Academics</td>
<td>*</td>
<td>*</td>
<td></td>
<td>***</td>
</tr>
<tr>
<td><strong>HID Spoofing</strong></td>
<td>White Hat Corporate espionage</td>
<td>**</td>
<td>***</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>Human Interface Device</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0-day</td>
<td>Government High-end corp espionage</td>
<td>****</td>
<td>****</td>
<td>****</td>
<td>*</td>
</tr>
</tbody>
</table>
Description of non-physical threats

- Social engineering
- **Software vulnerabilities**
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)

Software vulnerabilities

- Software *always* has errors
- Many software bugs are exploitable
  - e.g., buffer overrun -> overwrite variables
- Some errors are easy to fix (or prevent)
  - e.g., range checking in typesafe languages
- Other errors need a big, expensive, invasive changes or system redesign
  - change of protocol, new file format, authorization changes
- 0-day vulnerabilities are new errors, that are not yet known or fixed in the software
  - But even “old” vulnerabilities still useful
    (Android phones, embedded/IoT firmware)
Software Vulnerabilities

- **Buffer / heap / stack overflows**
  - Overwriting memory locations adjacent to a buffer

- **Unvalidated input, including SQL injections**
  - Unvalidated input causing unexpected behaviour of software

- **Race conditions**
  - Changes to the order of several events cause a change in behaviour

- **Insecure file operations**
  - Incorrect assumptions about ownership, location or attributes of a file

- **Side-channel leakage**
  - Unprotected implementation leading to a leakage of secret information via side channels, e.g., time, power, sound...

- **Weaknesses in the implementation of access control**
  - Authentication and authorization flaws
Software Vulnerabilities

Cisco Patches ‘Critical’ ASA IKE Buffer Overflow Vulnerability

- The algorithm for re-assembling IKE payloads fragmented with the Cisco fragmentation protocol contained a flaw that allowed a heap buffer to be overflowed with attacker-controlled data.
- Attackers could use this vulnerability to execute arbitrary code on affected devices.

Source: TripWire

Buffer overflow attacks target Facebook and MySpace

Buffer overflow attacks are targeting the Facebook and MySpace social networking sites.

Source: Computer Weekly

SQLI HALL-OF-SHAME

Welcome to the SQL Injection Hall-of-Shame

In this day and age it’s ridiculous how frequently large organizations are falling prey to SQL Injection (SQLI) which is almost totally preventable as I’ve tell people all the time as part of my day job at Parasoft and written previously.

Note that this is a work in progress. If I’ve missed something you’re aware of please let me know in the comments at the bottom of the page or on Twitter.

Source: The Code Curmudgeon
# Software Vulnerabilities

- **OWASP Top 10 - 10 Most Critical Web Application Security Risks**

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Vulnerability Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Injection</td>
<td>Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker’s hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.</td>
</tr>
<tr>
<td>A2</td>
<td>Broken Authentication and Session Management</td>
<td>Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users’ identities.</td>
</tr>
<tr>
<td>A3</td>
<td>Cross-Site Scripting (XSS)</td>
<td>XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim’s browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.</td>
</tr>
<tr>
<td>A4</td>
<td>Insecure Direct Object References</td>
<td>A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.</td>
</tr>
<tr>
<td>A5</td>
<td>Security Misconfiguration</td>
<td>Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.</td>
</tr>
<tr>
<td>A6</td>
<td>Sensitive Data Exposure</td>
<td>Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identify theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.</td>
</tr>
<tr>
<td>A7</td>
<td>Missing Function Level Access Control</td>
<td>Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.</td>
</tr>
<tr>
<td>A8</td>
<td>Cross-Site Request Forgery (CSRF)</td>
<td>A CSRF attack forces a logged-on victim’s browser to send a forged HTTP request, including the victim’s session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim’s browser to generate requests the vulnerable application thinks are legitimate requests from the victim.</td>
</tr>
<tr>
<td>A9</td>
<td>Using Components with Known Vulnerabilities</td>
<td>Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.</td>
</tr>
<tr>
<td>A10</td>
<td>Unvalidated Redirects and Forwards</td>
<td>Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.</td>
</tr>
</tbody>
</table>

Source: [OWASP](https://owasp.org)
Description of non-physical threats

● Social engineering
● Software vulnerabilities
● Distributed Denial of Service
● Side-channels
● Malicious Software (Malware)

Distributed Denial of Service - (D)DoS

● Make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of an Internet-connected host
● Typically accomplished by flooding the target with valid but superfluous requests attempting to overload the system and prevent some or all legitimate requests from being fulfilled
Distributed Denial-of-Service
DDoS Live

Source: Digital Attack Map
Mirai – The 150’000 IoT-Camera Botnet

- 2016-09-20: KrebsOnSecurity (KOS) DDoS’ed
  - Record breaking traffic: 620 Gbps
    (previous record: 363 Gbps)
  - Akamai had to drop DoS protection for KOS
  - Later: KOS protected by Google’s Project Shield
- 2016-09-22: OVH hit by 1 Tbps (!) traffic
- 2016-10-21: Dyn DNS provider targeted
  - Massive Internet outage
  - Affects many large companies (Amazon, GitHub, Netflix, NYT, Spotify, Twitter, WIRED, …)
- 2016-11-04: Liberia knocked offline
- 2016-11-30: 900k German Telekom routers knocked offline by expanding Mirai worm
- To-be-continued …
How-To Botnet in 2017

Source: Shodan.io
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- **Side-channels**
- Malicious Software (Malware)

Side-channel Attacks

- Attacks that extract secret information based on physical or temporal properties of an implementation
- Targets usually cryptographic software/hardware
- Forms:
  - Cache behaviour analysis
  - Timing analysis
  - Power analysis
  - Electromagnetic analysis
  - Acoustic analysis
  - Error inducing (Rowhammer)
  - …
Timing Attacks

Remote Timing Attacks are Practical

David Brumley
Stanford University
dbrumley@cs.stanford.edu

Dan Boneh
Stanford University
dabo@cs.stanford.edu

Abstract

Timing attacks are usually used to attack weak computing devices such as smartcards. We show that timing attacks apply to general software systems. Specifically, we devise a timing attack against OpenSSL. Our experiments show that we can extract private keys from an OpenSSL-based web server running on a machine in the local network. Our results demonstrate that timing attacks against network servers are practical and therefore security systems should defend against them.

The attacking machine and the server were in different buildings with three routers and multiple switches between them. With this setup we were able to extract the SSL private key from common SSL applications such as a web server (Apache+mod.SSL) and a SSL-tunnel.

Interprocess. We successfully mounted the attack between two processes running on the same machine. A hosting center that hosts two domains on the same machine might give management access to the admins of each domain. Since both domain are hosted on the same machine, one admin could use the attack to extract the secret key belonging to the other domain.

Remote Timing Attacks Are Still Practical*

Billy Bob Brumley and Nicola Tuveri
Aalto University School of Science, Finland
{brumley,ntuveri}@cs.hut.fi

Abstract. For over two decades, timing attacks have been an active area of research within applied cryptography. These attacks exploit cryptosystem or protocol implementations that do not run in constant time. When implementing an elliptic curve cryptosystem with a goal to provide side-channel resistance, the scalar multiplication routine is a critical component. In such instances, one attractive method often suggested in the literature is Montgomery’s ladder that performs a fixed sequence of curve and field operations. This paper describes a timing attack vulnerability in OpenSSL’s ladder implementation for curves over binary fields. We use this vulnerability to steal the private key of a TLS server where the server authenticates with ECDSA signatures. Using the timing of the exchanged messages, the messages themselves, and the signatures, we mount a lattice attack that recovers the private key. Finally, we describe and implement an effective countermeasure.
Acoustic Cryptanalysis

New attack steals e-mail decryption keys by capturing computer sounds

Scientists use smartphone to extract secret key of nearby PC running PGP app.

DAN GOODIN - 12/19/2013, 12:25 AM

Enlarge / In this photograph, (A) is a Lenovo ThinkPad T61 target; (B) is a Brüel&Kjær 4190 microphone capsule mounted on a Brüel&Kjær 2669 preamplifier held by a flexible arm, (C) is a Brüel&Kjær 5935 microphone power supply and amplifier, (D) is a National Instruments MyDAQ device with a 10 kHz RC low-pass filter cascaded with a 150 kHz RC high-pass filter on its A2D input, and (E) is a laptop computer performing the attack. Full key extraction is possible in this configuration, from a distance of 1 meter.

Source: Ars Technica
Side-Channel Attacks: Rowhammer

Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript

Daniel Gruss
Graz University of Technology, Austria
daniel.gruss@iaik.tugraz.at

Clemence Maurice
Technicolor, Rennes, France
Eurecom, Sophia-Antipolis, France
clementine.maurice@technicolor.com

Stefan Mangard
Graz University of Technology, Austria
stefan.mangard@tugraz.at

Abstract—As DRAM has been scaling to increase in density, the cells are less isolated from each other. Recent studies have found that repeated accesses to DRAM rows can cause random bit flips in an adjacent row, resulting in the so-called Rowhammer bug. This bug has already been exploited to gain root privileges and to evade a sandbox, showing the severity of faulting single bits for security. However, these exploits are written in native code and use special instructions to flush data from the cache.

In this paper we present Rowhammer.js, a JavaScript-based implementation of the Rowhammer attack. Our attack uses an eviction strategy found by a generic algorithm that improves the eviction rate compared to existing eviction strategies from 95.2% to 99.99%. Rowhammer.js is the first remote software-induced hardware-fault attack. In contrast to other fault attacks it does not require physical access to the machine, or the execution of native code or access to special instructions. As JavaScript-based fault attacks can be performed on millions of users stealthily and simultaneously, we propose countermeasures that can be implemented immediately.

Rowhammer.js Is the Most Ingenious Hack I've Ever Seen

This JavaScript exploit lets your browser mess with computer memory in a way that shouldn't be possible.

Source: Motherboard
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)

Malicious Software (Malware)

- Malware refers to software programs designed to damage or do other unwanted actions on a computer system.
- These actions can be visible or be hidden, depending on the purpose of the malware.
Clicker-time - #2

What is the most common type of malware?
Types of Malicious Software Affecting Us

- **57%** Viruses
- **26%** Misc. Trojans
- **7%** Trojan downloaders and droppers
- **4%** Misc potentially unwanted software
- **3%** Adware
- **3%** Exploits
- **2%** Worms
- **2%** Password stealers and monitoring tools
- **1%** Backdoors
- **0.01%** Spyware

Source: ITProPortal
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)
  - **Viruses**
    - Hidden inside programs / files
    - Produces copies of itself which are inserted into other programs / files
    - Passive spreading: Requires user actions for distribution to other systems
  - Worms
  - Trojans
  - Rootkit
  - Ransomware
  - Backdoors
  - Nation-state malware
I LOVE YOU

 Origin: Philippines

 No. of Infected Computers?
 Approximately 500,000

 When May 2000

 Reported Damages:
 $15 Billion

 What?
 An email virus. Specifically, the email attachment labelled “I Love You” was what caused the real harm.

 How it Worked:
 Once run, it overwrote existing system files with copies of itself. Also, it downloaded a file called WIN-BUGSFIX.EXE from the web and executed it. Rather than fix bugs, this programme e-mailed sensitive information from the victim’s computer to the hacker’s e-mail address.

 Source: ITProPortal
Virus Venn Diagram

Source: xkcd
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)
  - Viruses
  - **Worms**
  - Trojans
  - Rootkit
  - Ransomware
  - Backdoors
  - Nation-state malware

Worms

- Standalone malware
- Active spreading:
  - Transmits itself over the network
  - Exploits software vulnerabilities
Morris Worm

- Created by Robert Tappan Morris at Cornell University in 1988
- One of the first Internet worms
- Intended goal: Map the existing Internet
  - Accidental side-effect: Computers could be infected multiple times, slowing them down until eventually becoming unusable
- Infected more than 6000 University, military and research center computers
- Exploited vulnerabilities in sendmail, finger and rsh/exec and weak passwords

Storm Worm

Origin:
Reportedly Russia but this is unconfirmed

What?
A backdoor Trojan horse that affected computers using Microsoft operating systems. It was transmitted via email and it was named Storm Worm because one of the e-mail messages carrying the virus had as its subject "230 dead as storm batters Europe."

When
January 2007

No. of Infected Computers?
Around 10 million

Storm Worm (Storm Trojan)

How it Worked:
Once an email attachment was opened the Trojan implanted a service called wincom32. This passed data to other infected computers and all of the infected computers became bots i.e. a huge global network of computers controlled by Storm Trojan/Storm Worm.

Reported Damages:
Unknown

Source: ITProPortal
Conficker

Worm Infests Millions of Computers Worldwide

By JOHN MARKOFF  JAN. 22, 2009

A new digital plague has hit the Internet, infecting millions of personal and business computers in what seems to be the first step of a multistage attack. The world’s leading computer security experts do not yet know who programmed the infection, or what the next stage will be.

In recent weeks a worm, a malicious software program, has swept through corporate, educational and public computer networks around the world. Known as Conficker or Downadup, it is spread by a recently discovered Microsoft Windows vulnerability, by guessing network passwords and by hand-carried consumer gadgets like USB keys.

Experts say it is the worst infection since the Slammer worm exploded through the Internet in January 2003, and it may have infected as many as nine million personal computers around the world.

Source: New York Times
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)
  - Viruses
  - Worms
  - Trojans
  - Rootkit
  - Ransomware
  - Backdoors
  - Nation-state malware

Trojans

- Disguises itself as a useful program tricking victims into installing it
- Often combined with social engineering
- Does usually not propagate itself
- Usually has remote access capabilities
ZeuS

- High-tech Trojan used to create Botnets
- Sold as a kit for $3000 - $4000
- Offers several for-pay extensions, e.g., Jabber ($500) or VNC ($10’000)
- Employs advanced stealth techniques to avoid AV
- Encrypted peer-to-peer communication
- Features hardware-based licensing system
- Spread by
  - Drive-by-downloads
  - Phishing
- Application purpose:
  - Banking fraud (keylogging, form stealing)
  - Spread ransomware (CryptoLocker)
- Infected > 3.6 million PCs in the US by 2009

Source: Symantec
Fileless Malware

- Trades persistence for stealth:
  - Does not store files on disk
  - Does not survive reboots (often not a problem because computers are rarely rebooted nowadays)
  - Undetectable by AV scanners

- Usage scenarios:
  - Load other malware (e.g., rootkits, ransomware)
  - Perform click-fraud
  - Send spam
  - …

Source: Wired
Registry Malware

- Advanced fileless malware
- Achieves stealthiness and persistence
- Infects Windows registry
- Undetectable by AV scanners

Source: Symantec
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)
  - Viruses
  - Worms
  - Trojans
  - Rootkit
  - Ransomware
  - Backdoors
  - Nation-state malware

Rootkits

- Advanced malware concealing itself to avoid detection
- Enables continued privileged access
- There are types for user / kernel mode, hypervisors, firmware, and hardware
- Removal usually (very) difficult
Firmware Rootkits – The Attacker’s Holy Grail

- Most powerful and sophisticated technique
- Hard to develop and deploy
- Survives reboots, system updates and reinstallations
- Has hidden storage to keep valuable data:
  - Exfiltrated files
  - Full-disk encryption keys / passwords
- Almost impossible to detect / remove:
  - Undetectable by AV scanners
  - Manually checking for altered firmware is difficult and requires expert knowledge
  - Replacing the infected device is often the only “solution” to get rid of the malware

Source: Wired
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)
  - Viruses
  - Worms
  - Trojans
  - Rootkit
  - Ransomware
  - Backdoors
  - Nation-state malware

Ransomware

- Malware that locks your computer until victim has paid a ransom (usually in Bitcoin)
- Newer versions (starting with CryptoLocker in 2013) encrypt data using public-key cryptography
Ransomware

For example: In 2014 CryptoLocker extorted about $23 million from victims (according to an estimation by Symantec)

Source: Wired
Ransomware – More of the Same

- **CryptoDefense:**
  - Enforced payments over Tor
  - Handed out Tor installation guides to victims

- **CryptoWall:**
  - Also encrypts your external drives (i.e., backups)
  - Has an affiliate program giving criminals a cut of the profit if they help spread the word

- **CTB-Locker:**
  - CTB: Curve-Tor-Bitcoin
  - Uses Elliptic curve cryptography
  - Command servers on Tor
  - Payments via Bitcoin
  - Has an affiliates program as well

- **Soon:** Ransomware using **Zcash**?

Source: Wired
Ransomware – No One is Spared

RANSOMWARE TURNS TO BIG TARGETS—WITH EVEN BIGGER FALLOUT

Source: Wired

Source: Talk by Jeremiah Grossman at RSAConf'17
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)
  - Viruses
  - Worms
  - Trojans
  - Rootkit
  - Ransomware
  - Backdoors
  - Nation-state malware

Definition of a backdoor by Zdziarski:

“A backdoor is a component of a security boundary mechanism, in which the component is active on a computer system without consent of the computer’s owner, performs functions that subvert purposes disclosed to the computer’s owner, and is under the control of an undisclosed actor.”
Backdoors

How to test whether a technology fulfils the definition:

- **Intent:** “Does the mechanism behave in a way that subverts purposes as disclosed to the computer owner?”
- **Consent:** “Is the mechanism, or are subcomponents of the mechanism, active on a computer without the consent of the computer’s owner?”
- **Access:** “Is the mechanism under the control of an undisclosed actor?”
Clipper Chip

The idea is to give the Government means to override other people's codes, according to a concept called "key escrow." Employing normal cryptography, two parties can communicate in total privacy, with both of them using a digital "key" to encrypt and decipher the conversation or message. A potential eavesdropper has no key and therefore cannot understand the conversation or read the data transmission. But with Clipper, an additional key -- created at the time the equipment is manufactured -- is held by the Government in escrow. With a court-approved wiretap, an agency like the F.B.I. could listen in. By adding Clipper chips to telephones, we could have a system that assures communications will be private -- from everybody but the Government.

Source: The New York Times

- The Clipper chip satisfies the three requirements Intent, Consent, and Access of Zdziarski’s backdoor definition
- Discussion on key escrow is more relevant than ever, see, e.g., Apple vs. FBI in 2016
Dual-EC DRBG

Dual EC: A Standardized Back Door

Daniel J. Bernstein¹,², Tanja Lange³, and Ruben Niederhagen¹

¹ Department of Mathematics and Computer Science
Technische Universität Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
tanja@hyperelliptic.org, ruben@polycephaly.org
² Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607–7045, USA
djb@cr.yp.to

Abstract. Dual EC is an algorithm to compute pseudorandom numbers starting from some random input. Dual EC was standardized by NIST, ANSI, and ISO among other algorithms to generate pseudorandom numbers. For a long time this algorithm was considered suspicious – the entity designing the algorithm could have easily chosen the parameters in such a way that it can predict all outputs – and on top of that it is much slower than the alternatives and the numbers it provides are more biased, i.e., not random.

The Snowden revelations, and in particular reports on Project Bullrun and the SIGINT Enabling Project, have indicated that Dual EC was part of a systematic effort by NSA to subvert standards. This paper traces the history of Dual EC including some suspicious changes to the standard, explains how the back door works in real-life applications, and explores the standardization and patent ecosystem in which the standardized back door stayed under the radar.

On the Practical Exploitability of Dual EC in TLS Implementations

Stephen Checkoway¹, Matthew Fredrikson², Ruben Niederhagen³, Adam Everspaugh⁴, Matthew Green¹, Tanja Lange⁵, Thomas Ristenpart⁶, Daniel J. Bernstein⁷,⁸ Jake Maskiewicz⁷ and Hovav Shacham⁹

¹ Johns Hopkins University, ² University of Wisconsin, ³ Technische Universität Eindhoven,
⁴ University of Illinois at Chicago, ⁵ UC San Diego

Abstract

This paper analyzes the actual cost of attacking TLS implementations that use NIST’s Dual EC pseudorandom number generator, assuming that the attacker generated the constants used in Dual EC. It has been known for several years that an attacker generating these constants and seeing a long enough stretch of Dual EC output bits can predict all future outputs; but TLS does not naturally provide a long enough stretch of output bits, and the cost of an attack turns out to depend heavily on choices made in implementing the RNG and on choices made in implementing other parts of TLS.

Specifically, this paper investigates OpenSSL–FIPS, Windows’s SChannel, and the C/C++ and Java versions of the RSA BSAFE library. This paper shows that Dual EC exploitability is fragile, and in particular is stopped by an outright bug in the certified Dual EC implementation in OpenSSL. On the other hand, this paper also shows that Dual EC exploitability benefits from a modification made to the Dual EC standard in 2007; from several attack optimizations introduced here; and from various proposed TLS extensions, one of which is implemented in BSAFE, though disabled in the version we obtained and studied. The paper’s attacks are implemented; benchmarked; tested against libraries modified to use new Dual EC constants; and verified to successfully recover TLS plaintext.

The documents also make specific reference to a set of pseudorandom number generator (PRNG) algorithms adopted as part of the National Institute of Standards and Technology (NIST) Special Publication 800-90 [21] in 2006, and also standardized as part of ISO 18031 [15]. These standards include an algorithm called the Dual Elliptic Curve Deterministic Random Bit Generator (Dual EC). As a result of these revelations, NIST reopened the public comment period for SP 800-90.

Known weaknesses in Dual EC. Long before 2013, Dual EC had been identified by the security community as biased [8, 27], extremely slow, and backdoorable.

SP 800-90 had already noted that “elliptic curve arithmetic” makes Dual EC generate “pseudorandom bits more slowly than the other DRBG mechanisms in this Recommendation” [21, p. 177] but had claimed that the Dual EC design “allows for certain performance-enhancing possibilities.” In fact, Dual EC with all known optimizations is two orders of magnitude slower than the other PRNGs, because it uses scalar multiplications on an elliptic curve where the other PRNGs use a hash function or cipher call.

The back door is a less obvious issue, first brought to public attention by Shumow and Ferguson [28] in 2007. What Shumow and Ferguson showed was that an attacker specifying Dual EC, and inspecting some Dual EC output bits from an unknown seed, had the power to predict all

Source: projectbullrun.org

NSA paid $10 million to put its backdoor in RSA encryption, according to Reuters report

by Russell Brandon | @russellbrandon | Dec 20, 2013, 4:34pm EST

Source: The Verge

Source: dualec.org
What do you think about backdoors?
Description of non-physical threats

- Social engineering
- Software vulnerabilities
- Distributed Denial of Service
- Side-channels
- Malicious Software (Malware)
  - Viruses
  - Worms
  - Trojans
  - Rootkit
  - Ransomware
  - Backdoors
  - Nation-state malware

Nation-state Malware

- Highly advanced malware
- Used for targeted attacks (often espionage, but sometimes physical)
- For infection commonly uses a combination of
  - Social engineering
  - Zero-day exploits
  - Other advanced attack vectors (e.g., hash function collision attacks)
Air-Gap Bridging Malware

Attack outline:

- Plant malware on air-gapped system (e.g., by paying insider to plug in an infected SD card / USB drive)
- Malware encodes data into signal using blinking LEDs
- Position drone with a high-res camera in front of a window to pick up the signal and exfiltrate data (at up to 4000 bps)

Source: Wired
Stuxnet

- Highly advanced malware
- Used for targeted sabotage of Iran’s nuclear program
- Supposedly developed by an American-Israeli team
- Exploited four zero-day exploits in Microsoft Windows
- Accidentally spread beyond its intended target due to a programming error
Stuxnet

HOW STUXNET WORKED

1. infection
Stuxnet enters a system via a USB stick and proceeds to infect all machines running Microsoft Windows. By brandishing a digital certificate that seems to show that it comes from a reliable company, the worm is able to evade automated-detection systems.

2. search
Stuxnet then checks whether a given machine is part of the targeted industrial control system made by Siemens. Such systems are deployed in Iran to run high-speed centrifuges that help to enrich nuclear fuel.

3. update
If the system isn’t a target, Stuxnet does nothing; if it is, the worm attempts to access the Internet and download a more recent version of itself.

4. compromise
The worm then compromises the target system’s logic controllers, exploiting “zero day” vulnerabilities—software weaknesses that haven’t been identified by security experts.

5. control
In the beginning, Stuxnet spies on the operations of the targeted system. Then it uses the information it has gathered to take control of the centrifuges, making them spin themselves to failure.

6. deceive and destroy
Meanwhile, it provides false feedback to outside controllers, ensuring that they won’t know what’s going wrong until it’s too late to do anything about it.

Source: IEEE Spectrum
Flame

- Highly advanced self-modifying malware
- Used for **targeted espionage** (mostly in the Middle East)
- Uncommonly large: 20MB
- Supported 5 different encryption methods
- Contained an entire SQLite database (!)
- Used two exploits previously known from Stuxnet
- Signed by a fraudulent Microsoft certificate (created by an MD5 collision)

Soon: Flame 2 - The Revenge ([http://shattered.io/](http://shattered.io/))?
OUR DEVICES ARE NOW 100% SECURE.

I TURNED THEM ALL OFF.

HOW DID YOU DO THAT?
Clicker-time - #4

How to target a specific person?
Conclusion

- Random attacks mostly use viruses, phishing, commodity malware
  - More concerned by the number of infections
  - Don’t care who is infected
- Targeted attacks use social engineering, spear phishing, specialty malware
  - Social engineering to place malware at strategic places
  - DDoS to silence service / institution / person
- Different attack vectors often require different defenses