Introduction

Mustakimur R. Khandaker

Syllabus

Instructor: Mustakimur Rahman Khandaker.
Office: 547 Boyd Graduate Studies Research Center.
Email: mrkhandaker@uga.edu

Office Hours: Monday and Thursday 12:00 PM - 01:00 PM or by appointment.

Lecture time and location:

Monday	10:20 am - 11:10 am
Boyd Grad Rsch Ctr	*0306*
Tuesday/Thursday	9:35 am - 10:50 am
Hardman Hall	*0101*

Prerequisite:

- CSCI 4730/6730 (Operating Systems)
  Or
- CSCI 4760/6760 (Computer Networks).

Instruction Policy:

- Course instructions will be (hopefully) in person.
  - Instruction policy depends on the UGA plan.
- Monday class will be mostly used as a lab section.
  - Showcase how to use a debugger, fuzzing tools, write compiler instrumentation and packet analysis.
  - Also, general discussion on project assignments.

Evaluation

Evaluation and Grading Policy:

- Quiz: 10% (5 quizzes).
  - Software attacks (week 5).
  - Software bug detection and defense mechanism (week 8).
  - Cryptography, web security, and mobile security (week 11).
  - Network security and protocols (week 14).
  - Hardware, IoT, CPS security; virtualization, and TEE (week 16).
- Homeworks: 20% (4 HW).
  - Software vulnerability analysis (week 5).
  - Shellcoding (week 7).
  - Hands-on cryptography (week 11).
  - Paper review on fuzzing (week 15).
- Projects: 40% (total 4).
  - Buffer overflow (week 5).
  - Return-to-Libc (week 8).
  - Format string (week 11).
  - Cross-Site Scripting (week 14).
- Midterm: 15%.
  - 100% (week 1-9).
- Final: 15%.
  - 40% (week 1-9).
  - 60% (week 9-17).

Letter Grade	Percentage
A	91 - 100%
A-	86 - 90%
B+	81 - 85%
B	76 - 80%
B-	71 - 75%
C+	66 - 70%
C	61 - 65%
C-	56 - 60%
D	51 - 55%
F	0 - 50%

Note: Late submission will receive a 3% penalty for each day (maximum 3 days allowed).

Course Plan

The general course objectives are to detect vulnerabilities, identify attacks, and protect proprietary systems and data across different cyber security fields.

Week 1: Syllabus, Security, and Principles.
Week 2-5: Software attacks.
Week 6-7: Software protection mechanism.
Week 7-9: Software bug detection.
Week 9-10: Cryptography.
Week 10-13: Web security and Mobile security.
Week 13-14: Network security and protocols.
Week 15: Hardware, IoT, and CPS security.
Week 16: Virtualization and TEE.
Week 17: Machine learning adversaries and Privacy.

Projects:
Assignments will be based on SEED Lab (SEED Project).
- Submit source code and malicious input.
- Submit a report detailing the process to exploit.

Homework/Project Policy:
- The homework will due at the start of the class on the specified day.
  - Submit digital copy.
- The project will due at Sunday 11:59 PM on the specific day.
  - Submit digital copy.
- Late submission will receive a 3% penalty for each day (maximum 3 days allowed).

Quiz/Exam Policy:
- The quiz will be held @ Monday class.
  - will be announced a week in advance.
- The final exam will be scheduled as UGA.
- The exam will be paper-based.
- Absences from the exam will not be permitted.

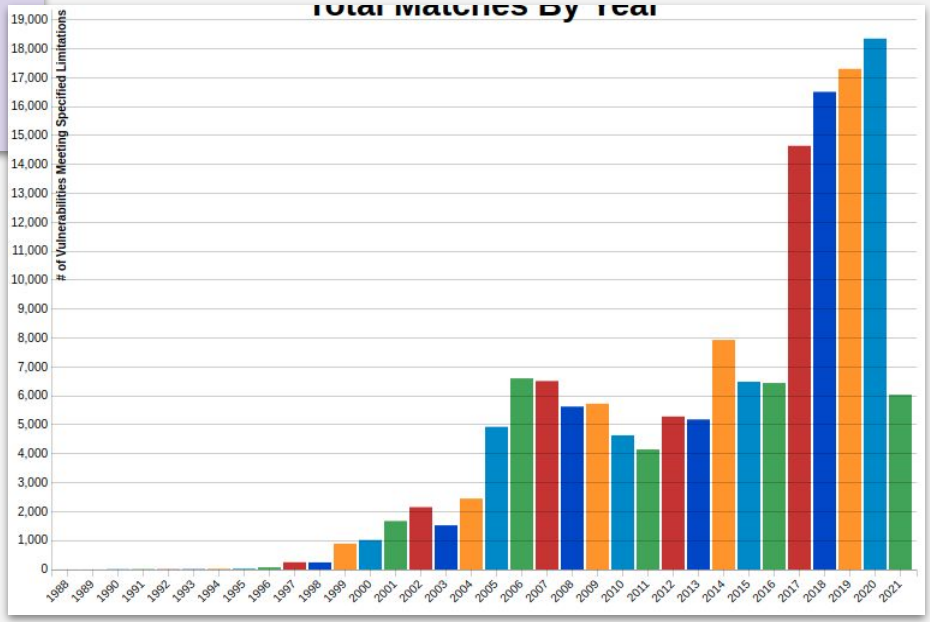
CyberSecurity

Problem:

- Vulnerable software and system.

Solution:

- Design secure system.
- Find out bugs.
- Patch software.
- Administer sensitive organization.



Impacted Industries/Organizations

- Industry
  - Software
    - Microsoft
    - Google
    - Apple
  - Hardware
    - Intel
    - HP
    - Qualcomm
- Open-source Community
  - Linux
  - Debian
  - Apache
- Government Organizations
  - NSA
  - NIH

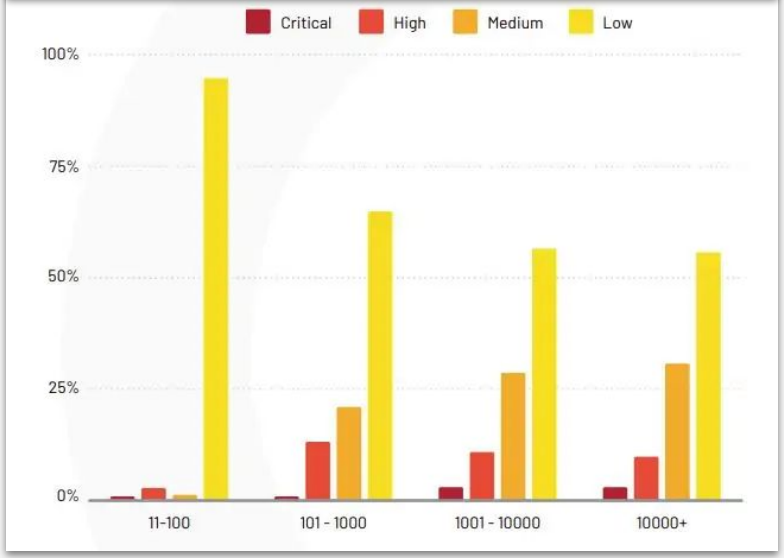
	Vendor Name	Number of Vulnerabilities
1	Microsoft	668
2	Google	609
3	Oracle	489
4	Adobe	441
5	Cisco	440
6	IBM	364
7	Debian	360
8	Cpanel	321
9	Redhat	257
10	Jenkins	254
11	Apple	229
12	Canonical	197
13	Fedoraproject	187
14	Qualcomm	171
15	Linux	170
16	Foxitsoftware	162
17	Opensuse	148
18	HP	129
19	Gitlab	119
20	Mozilla	118
21	Netapp	112
22	Apache	108
23	Intel	92
24	SAP	75
25	Magento	72
26	F5	63
27	Siemens	62
28	Dlink	61
29	Imagemagick	57
30	GNU	55
31	Schneider-electric	51
32	Zohocorp	49
33	Atlassian	49
34	Jetbrains	45
35	Cybozu	34
36	Mcafee	34
37	Juniper	34
38	Bestwebsoft	34
39	Moxa	34
40	Tcpdump	31

Year 2020

The NVD database holds 18,362 vulnerabilities published in 2020. This is a higher number than in previous years (17,382 in 2019 and 17,252 in 2018).

The oldest vulnerability discovered in 2020 was **21** years old.

Organizations with more than 100 staff see more high or critical-risk vulnerabilities



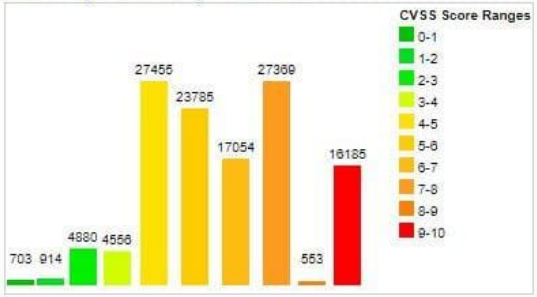
More than 13% of vulnerabilities have a critical score.

Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	703	0.60
1-2	914	0.70
2-3	4880	4.00
3-4	4556	3.70
4-5	27455	22.20
5-6	23785	19.30
6-7	17054	13.80
7-8	27369	22.20
8-9	553	0.40
9-10	16185	13.10
Total	123454	

Weighted Average CVSS Score: **6.6**

Vulnerability Distribution By CVSS Scores



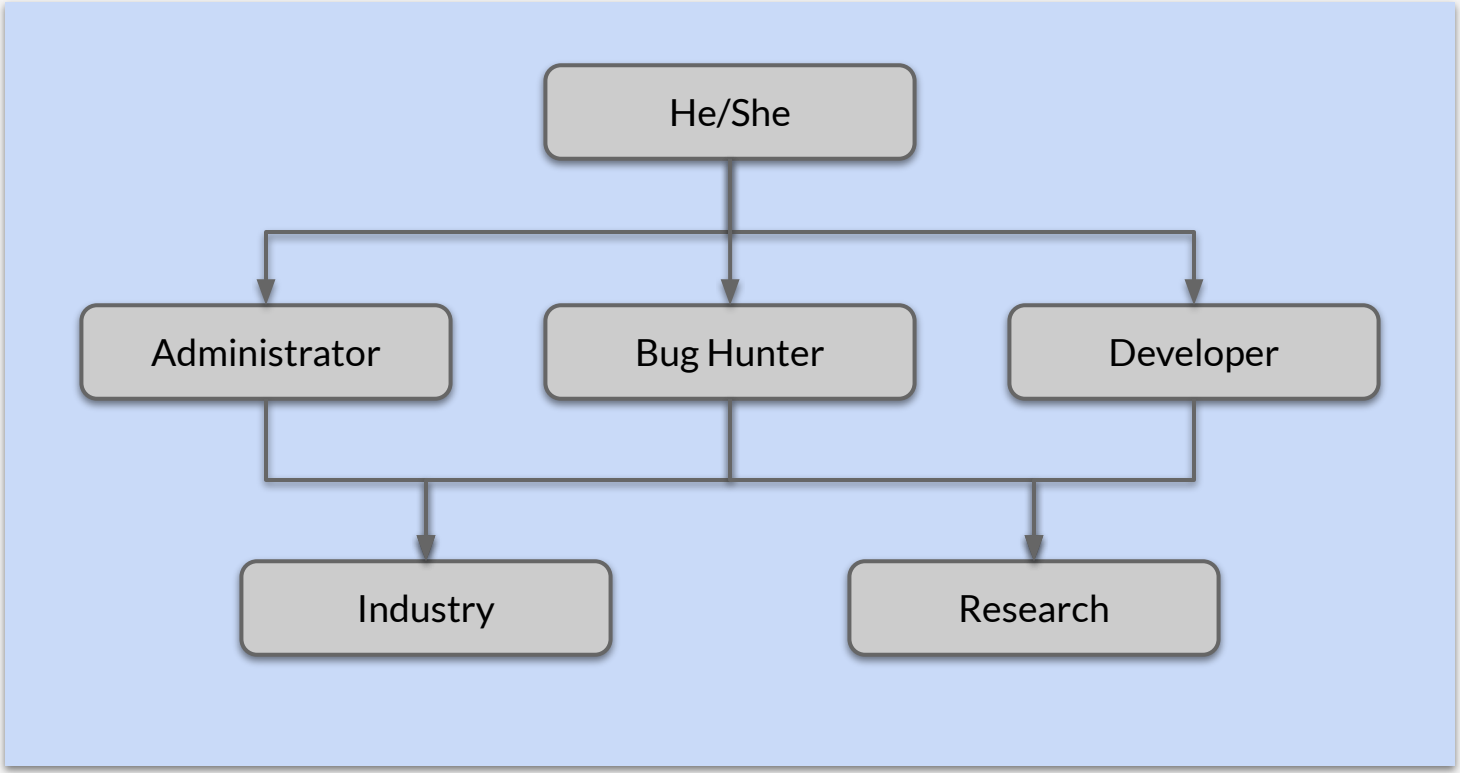
Reuters

IBM flags more cyber attacks on COVID vaccine infrastructure

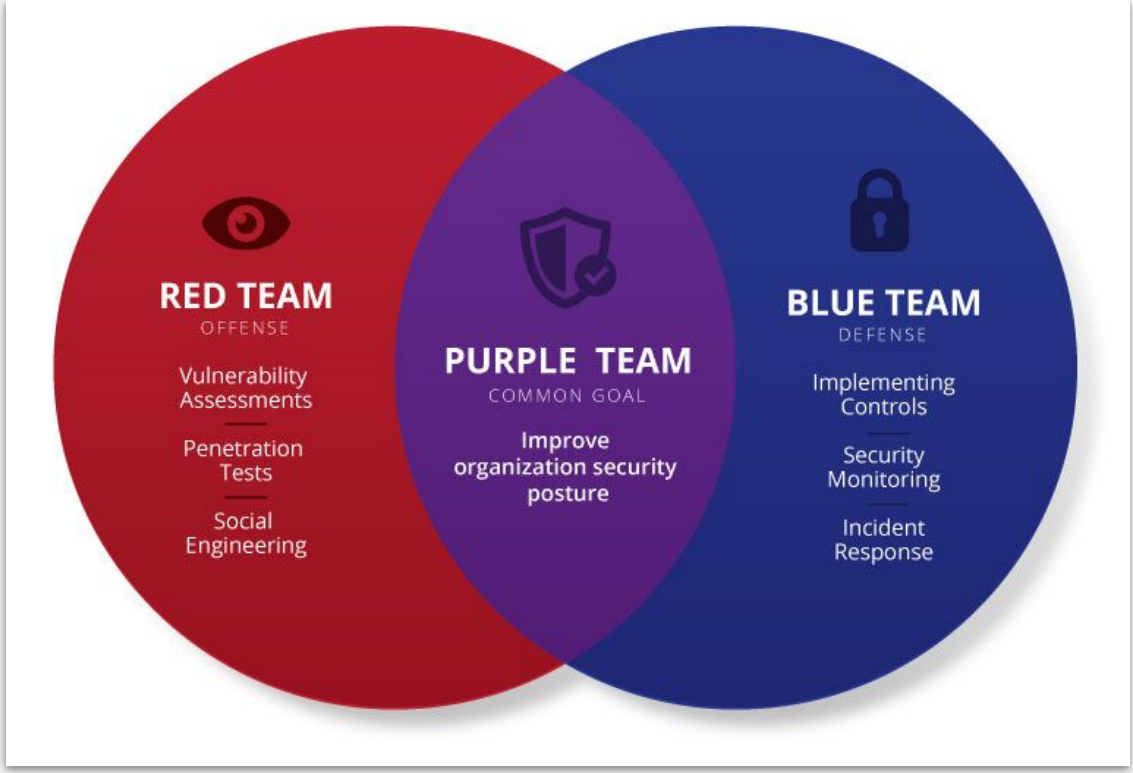
... Wednesday its cybersecurity unit has uncovered more digital attacks targeting the global COVID-19 vaccine supply chain since the issue was ...

2 weeks ago

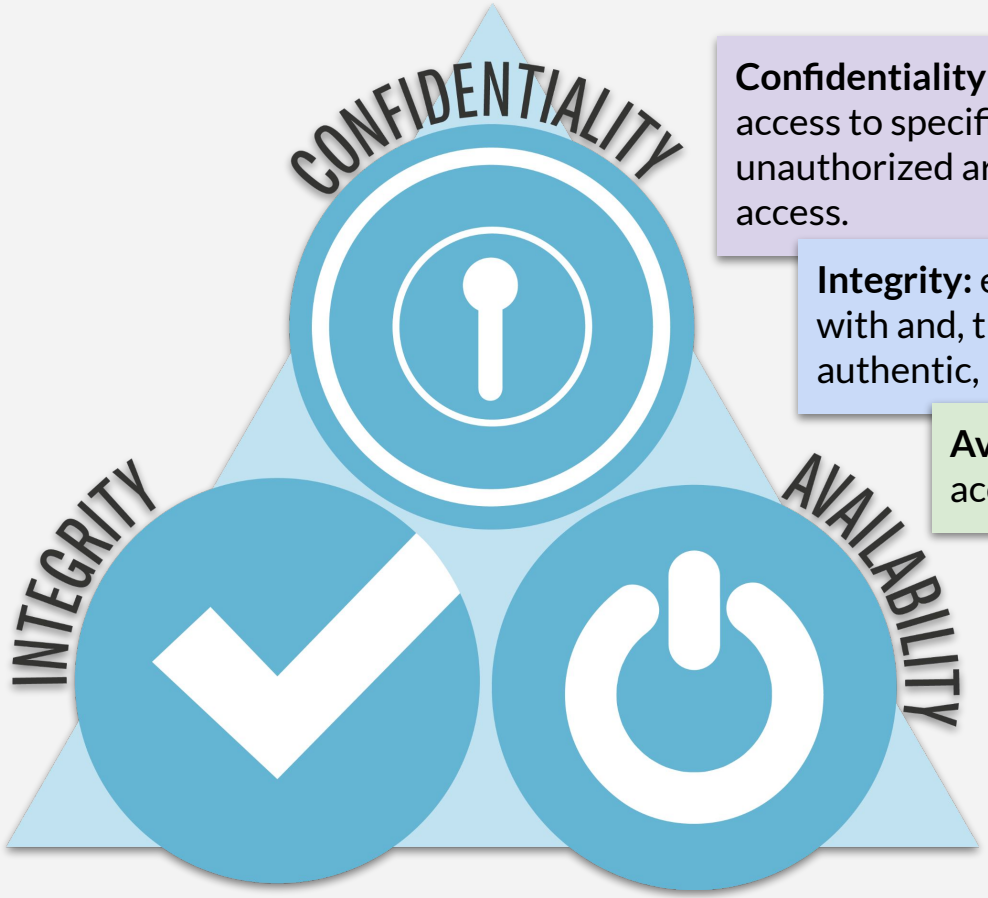
What you can be?



Organizational Structure



Principles

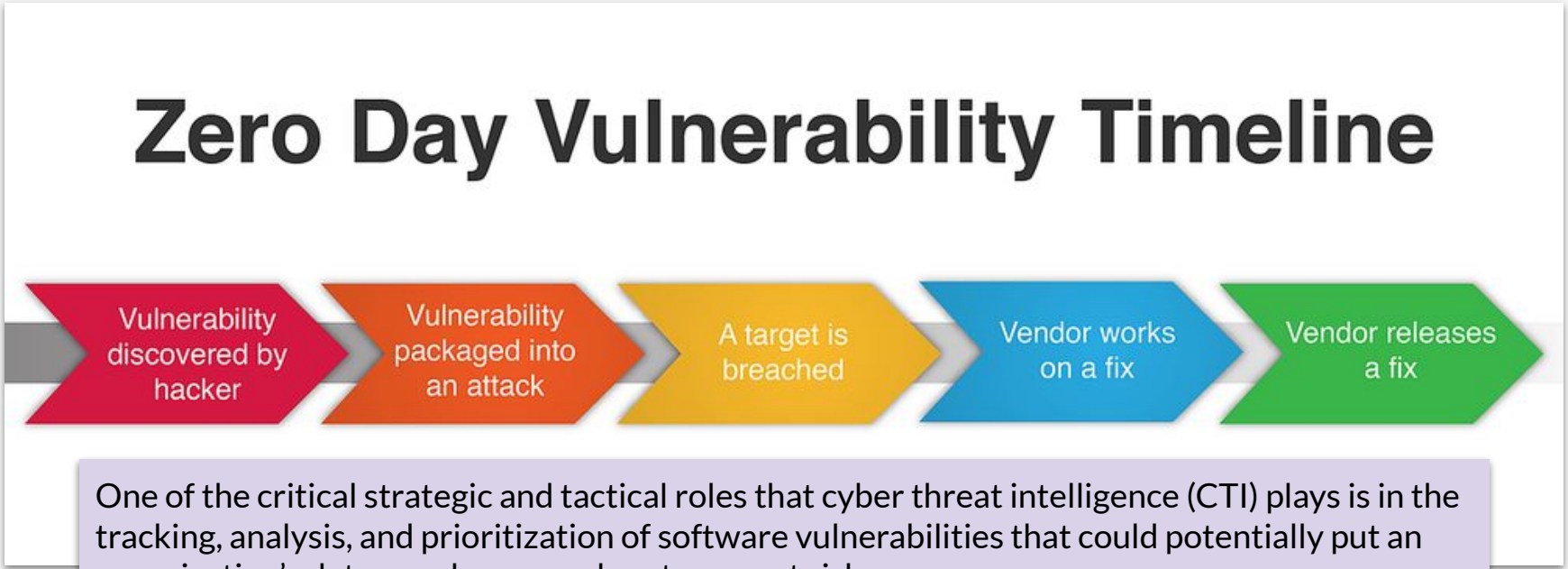


Confidentiality: only those who are authorized have access to specific assets and that those who are unauthorized are actively prevented from obtaining access.

Integrity: ensuring that data has not been tampered with and, therefore, can be trusted. It is correct, authentic, and reliable.

Availability: authorized users have timely, reliable access to resources when they are needed.

Zero Day



One of the critical strategic and tactical roles that cyber threat intelligence (CTI) plays is in the tracking, analysis, and prioritization of software vulnerabilities that could potentially put an organization's data, employees and customers at risk.

It Takes an Average **38** Days to Patch a Vulnerability.

Google's Project Zero adopts a approach, where the full details of the vulnerability are published after **90** days regardless of whether or not the organisation has published a patch.

Bugs (Sample)

```c
#include <stdio.h>

int main(int argc, char **argv)
{
    char buf[8];
    gets(buf);

    printf("%s\n", buf);
    return 0;
}
```

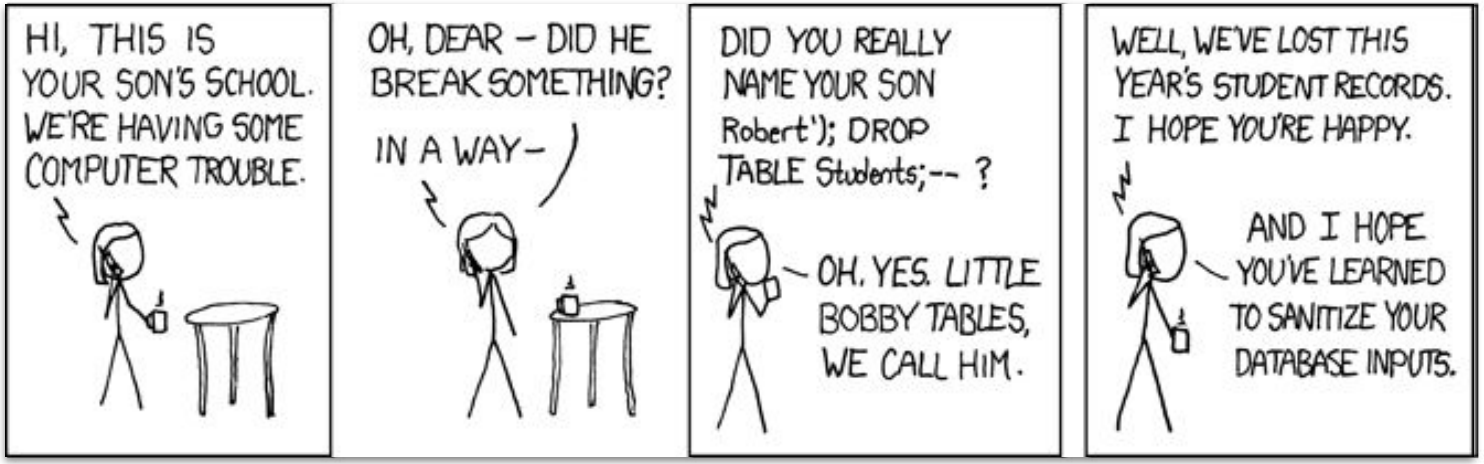
```c
nresp = packet_get_int();

if (nresp > 0) {
    response = xmalloc(nresp*sizeof(char*));

    for (i = 0; i < nresp; i++) {
        response[i] = packet_get_string(NULL);
    }
}
```

More Bugs

```
SELECT * FROM Users WHERE Username='$username' AND
Password='$password'
```



Source: http://xkcd.com

Epic Fail (Sony PS3 & Apple)

Random number generator for ECDSA:

```
int getRandomNumber ()
{
    return 4;
}
```

	Xbox	Wii	360	PS3
On-die bootROM	✓	✓	✓	✓
On-die key storage		✓	✓	
Public-key crypto	✓	✓	✓	✓
Chain of trust	✓		✓	✓
Per-console keys		✓	✓	✓
Signed executables	✓		✓	✓
Security coprocessor		✓		✓
Full media encryption and signing		✓		
Encrypted storage		✓		✓
Self-signed storage		✓		
Memory encryption/hashing			✓	
Hypervisor			✓	✓
User/kernelmode				✓
Anti-downgrade eFUSEs			✓	

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa,
        SSLBuffer signedParams, uint8_t *signature, UInt16 signatureLen)
{
    OSStatus        err;
    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```

Marketplace for Vulnerabilities

Option 1: bug bounty programs (many)

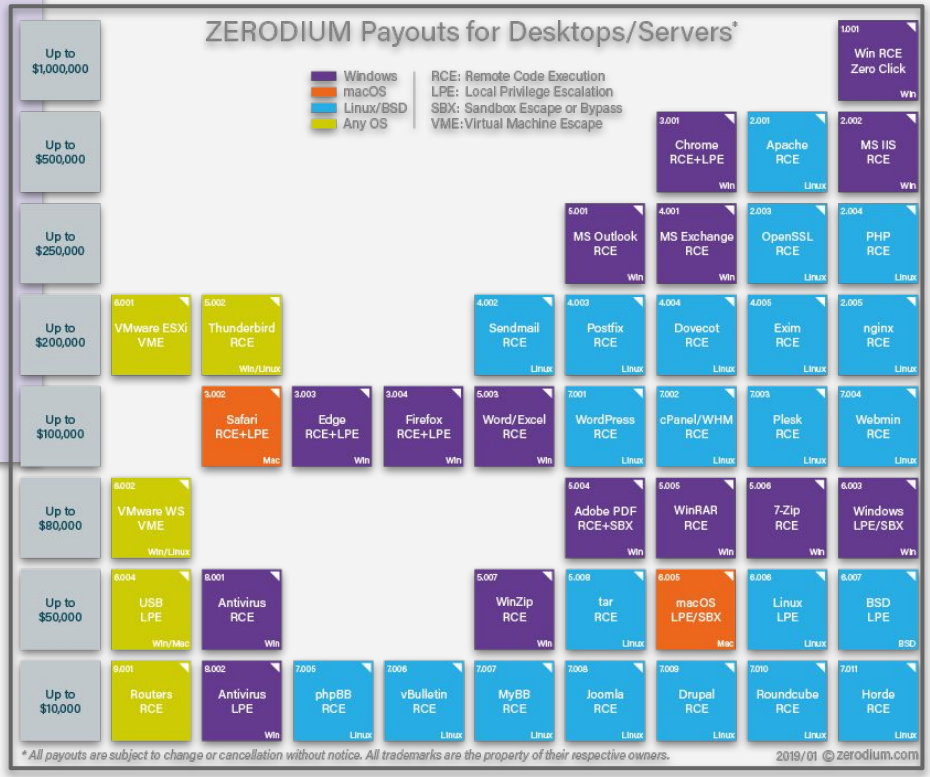
- Google Vulnerability Reward Program: up to 100K $
- Microsoft Bounty Program: up to 100K $
- Mozilla Bug Bounty program: 500$ - 3000$
- Pwn2Own competition: 15K $

Option 2:

- ZDI, iDefense.

Option 3: black market.

- Not really an option for ethical hackers.



Marketplace for Owned Machines

Pay-per-install (PPI) services:

- Install client's malware on owned machines for a fee.

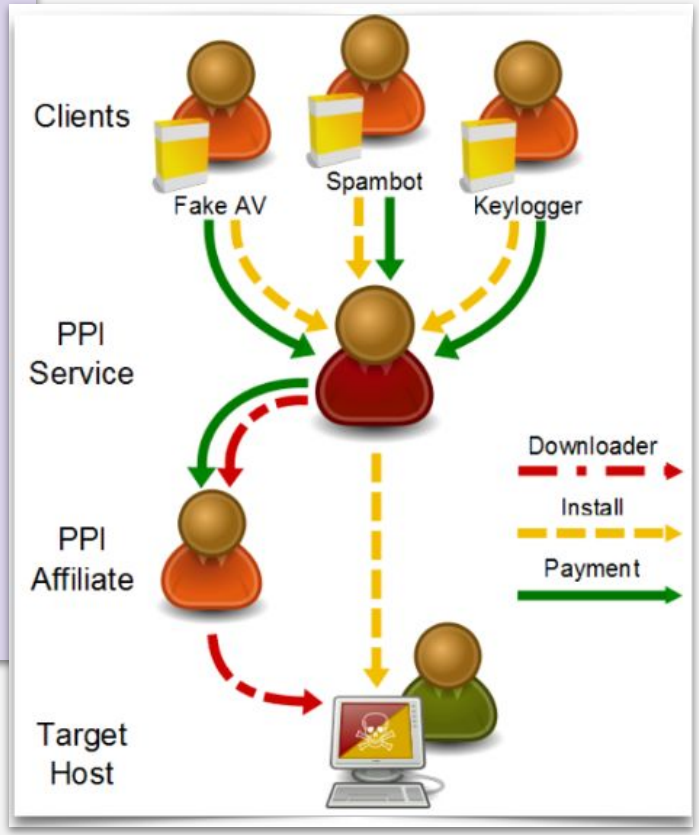
PPI operation:

- Own victim's machine.
- Download and install client's malware.
- Charge client.
  - US: 100-180$/1000 machines.
  - Asia: 7-8$/1000 machines.

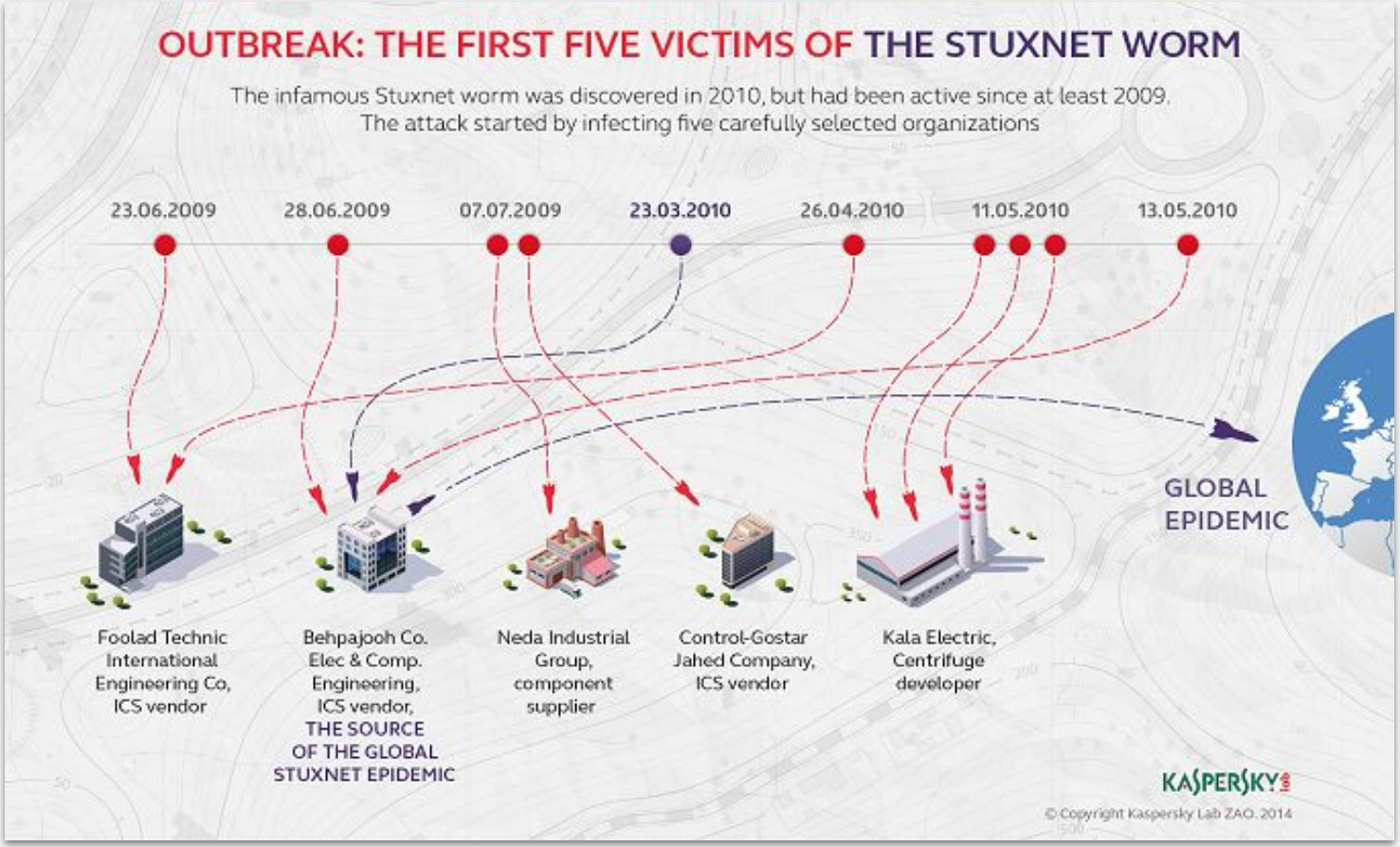
Steal IP Address and Bandwidth:

Use the IP address of infected machine of phone for

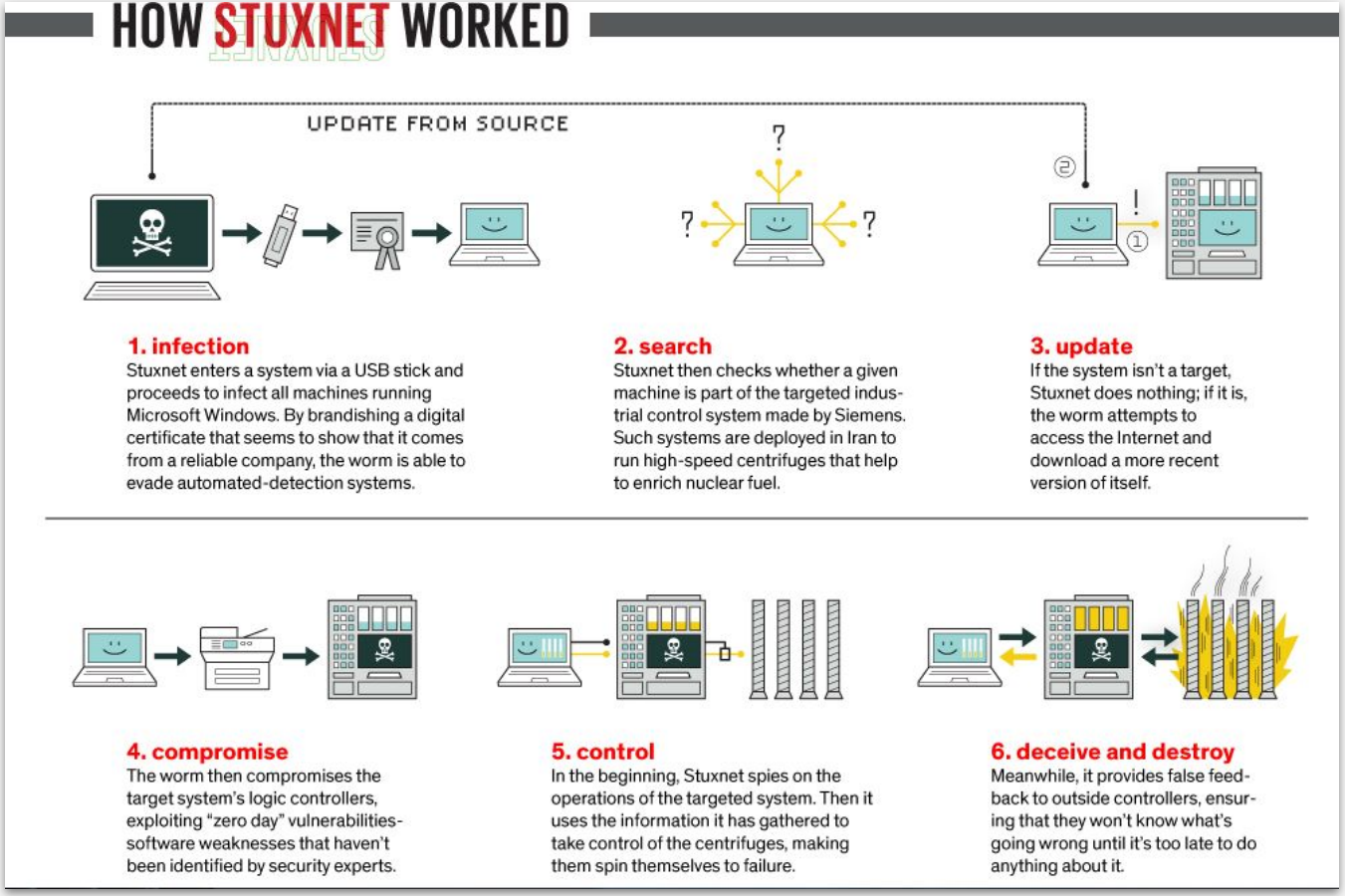
- Spam (e.g. the storm bonnet).
- Denial-of-service.
- Click fraud (clickbot.a).



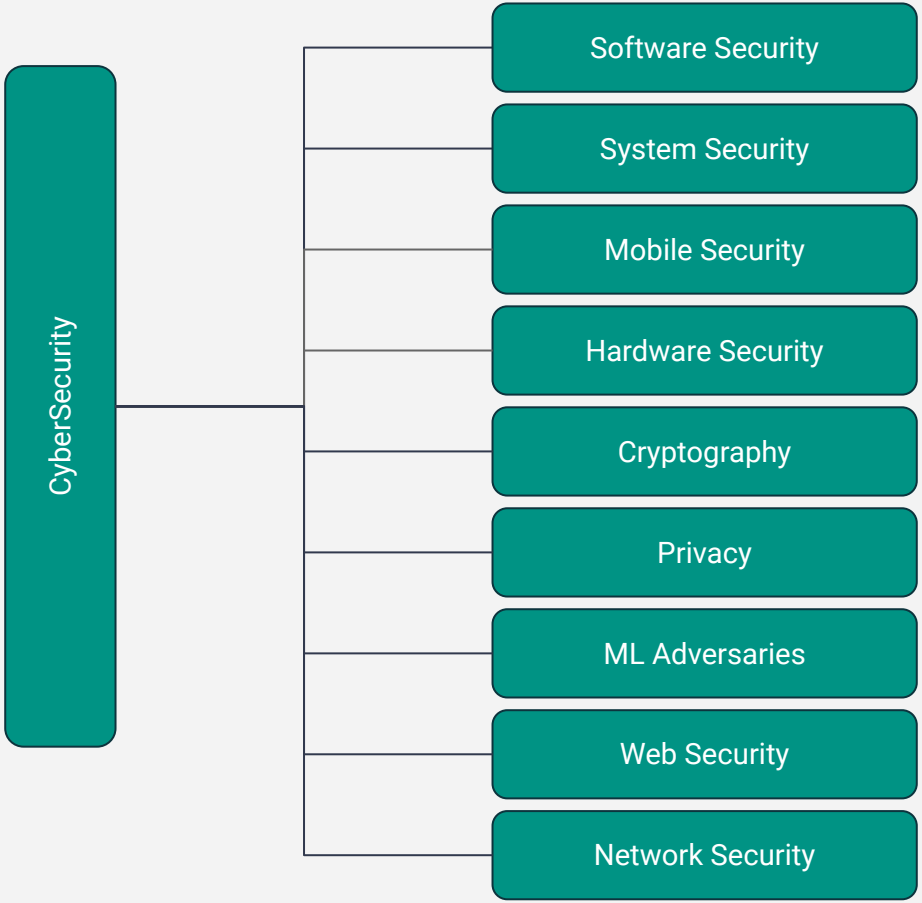
Stuxnet


OUTBREAK: THE FIRST FIVE VICTIMS OF THE STUXNET WORM
The infamous Stuxnet worm was discovered in 2010, but had been active since at least 2009.
The attack started by infecting five carefully selected organizations
23.06.2009
28.06.2009
07.07.2009
23.03.2010
26.04.2010
11.05.2010
13.05.2010
GLOBAL EPIDEMIC
Foolad Technic International Engineering Co, ICS vendor
Behpajooh Co. Elec & Comp. Engineering, ICS vendor, THE SOURCE OF THE GLOBAL STUXNET EPIDEMIC
Neda Industrial Group, component supplier
Control-Gostar Jahed Company, ICS vendor
Kala Electric, Centrifuge developer
KASPERSKY
© Copyright Kaspersky Lab ZAO, 2014

Continue ...


HOW STUXNET WORKED
UPDATE FROM SOURCE
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.

Areas


CyberSecurity
Software Security
System Security
Mobile Security
Hardware Security
Cryptography
Privacy
ML Adversaries
Web Security
Network Security

Summary

Relativity applies to physics, not ethics.