Special Topics in Security
ECE 5968

Engin Kirda
ek@ccs.neu.edu

Northeastern University

Admin News and Stuff

- Only a couple of more classes left
  - I'll be done with the Quiz 3 corrections in a couple of days
- There are a couple of more topics I would like to cover
- Last class – maybe we'll go over the material, as a quick recap (Q&A)

Botnets

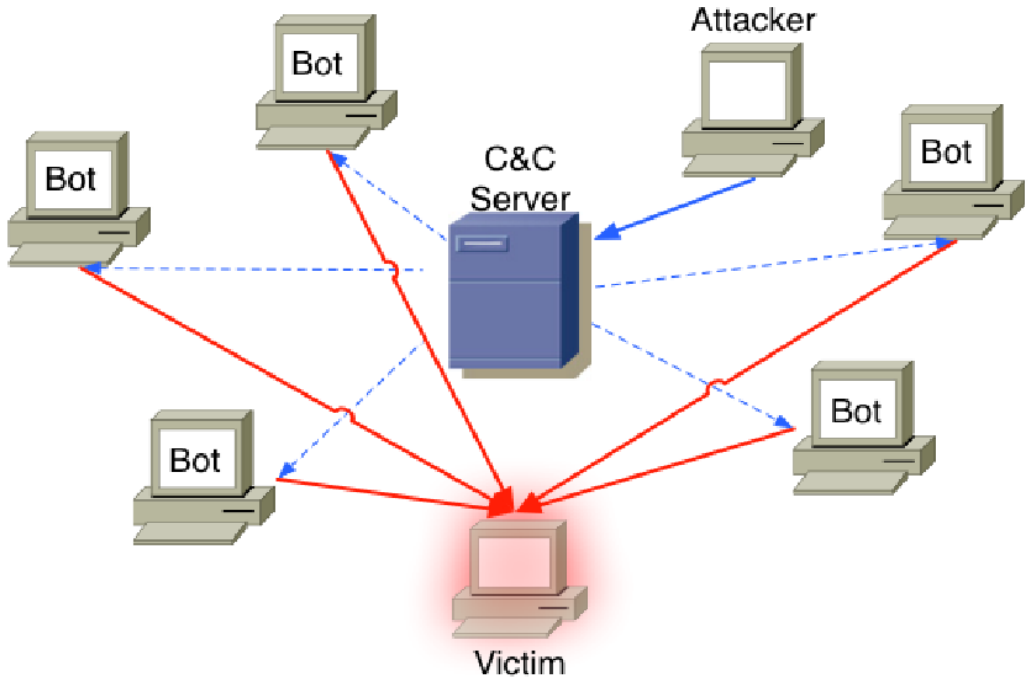
Bots

- A bot, a.k.a zombie or drone, is a compromised machine that can be controlled by an attacker remotely.

- Bots have three distinguishing features;
  - Remote control facility
  - Implementation of different commands
  - Spreading mechanism for propagation purposes

Botnets

- A botnet is a network that consists of several malicious bots that are controlled by a commander, called as *botmaster* or *botherder*



Historical Evolution of Botnets

- First bots were invented for benign use, worked in the IRC network;

The Jargon File, version 4.4.7:

> bot: n [common on IRC, MUD and among gamers; from "robot"]
>
> 1. An IRC or MUD user who is actually a program. On IRC, typically the robot provides some useful service.Examples are NickServ, which tries to prevent random users from adopting nicks already claimed by others, and MsgServ, which allows one to send asynchronous messages to be delivered when the recipient signs on.
>
> [...]

Historical Evolution of Botnets

- After a while, attackers abused the usage of IRC bots and waged IRC wars;
  - IRC wars were one of the first documented distributed denial of service attacks
- In late 1999, SANS researchers discovered remotely executable code on thousands of Windows machines.
  - First named as robots, later shortened to bots
  - The code was encrypted, therefore, how they worked could not be discovered by researchers
  - DDOS attack in February 2000
- Today, botnets are the most serious and dangerous type of malware

Simple Timeline of Botnets

Date	Name	Description
12/1993	EggDrop	Non-malicious IRC bot
04/1998	Gtbot	Malicious IRC bot based on MIRC
04/2002	Sdbot	Provided own IRC client
10/2002	Agobot	Robust, flexible, modular design
04/2003	Spybot	Extensive feature set based on Agobot
03/2004	Phatbot	P2P bot based on WASTE
03/2006	SpamThru	P2P bot
04/2006	Nugache	P2P bot
01/2007	Peacomm	P2P bot based on Kademlia
10/2007	Storm	Uses its own P2P network

The Botnet Threat

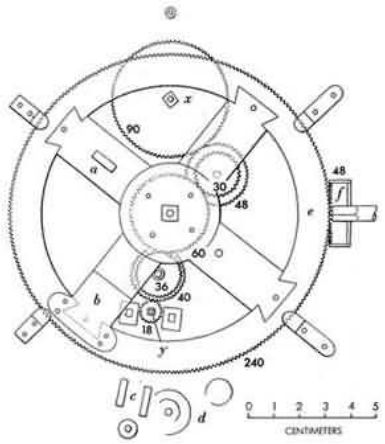
- Information dispersion
  - E-mail Spamming
  - Denial of Service attacks
- Information harvesting
  - Identity data
  - Financial data
  - Private data
  - E-mail address books
  - Any other type of data that may be present on the host of the victim

Botnet Characteristics

- Remote Control Facility
  - allows the attacker to have full control over the infected machines
- A wide range of supported commands
  - Allows the attacker to command bots for specific purposes
- Spreading mechanism for further propagation
  - e.g. exploiting vulnerabilities,
  - While remote control mechanism and commands differentiate bots from worms, they have similar mechanisms

Spreading Mechanisms

- The larger a bot is, the more effective it is...
- Propagation: finding vulnerable victims...
  - Random Scanning
  - Permutation Scanning
  - Hit-List Scanning
  - Combining techniques, e.g. Warhol worm



Command and Control Mechanism (C&C)

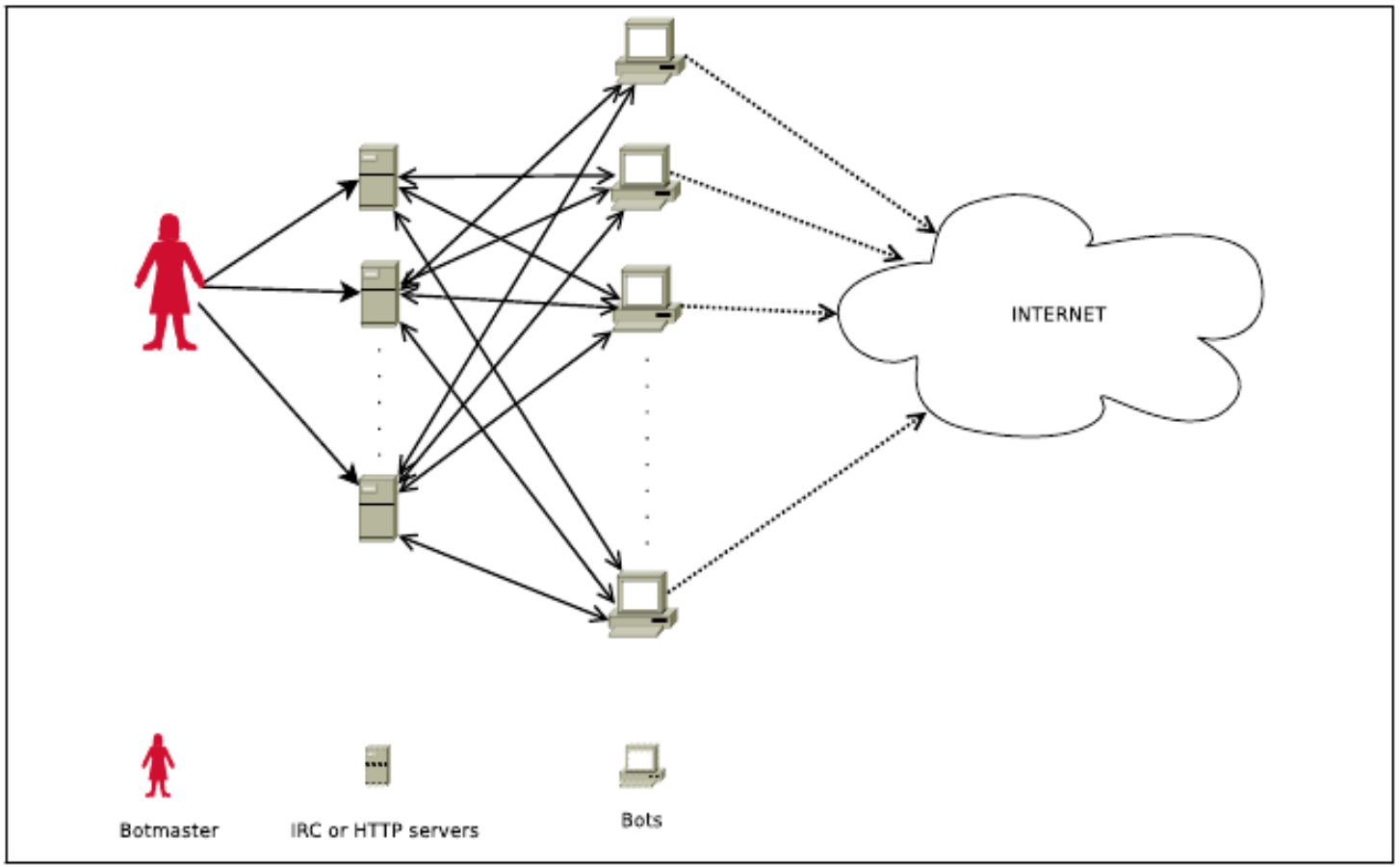
- The most distinguishing and powerful feature of Botnets
  - As long as, there is an ***update*** command defined for the bots, the botmaster can change the command set by updating her bots
- Thus
  - C&C brings a great flexibility to the activities that can be performed by bots
- However
  - C&C is the weakest link of the system

Command and Control Mechanism (C&C)

- Centralized Command and Control mechanisms
  - Push style C&C, e.g. IRC
  - Pull style C&C, e.g. HTTP

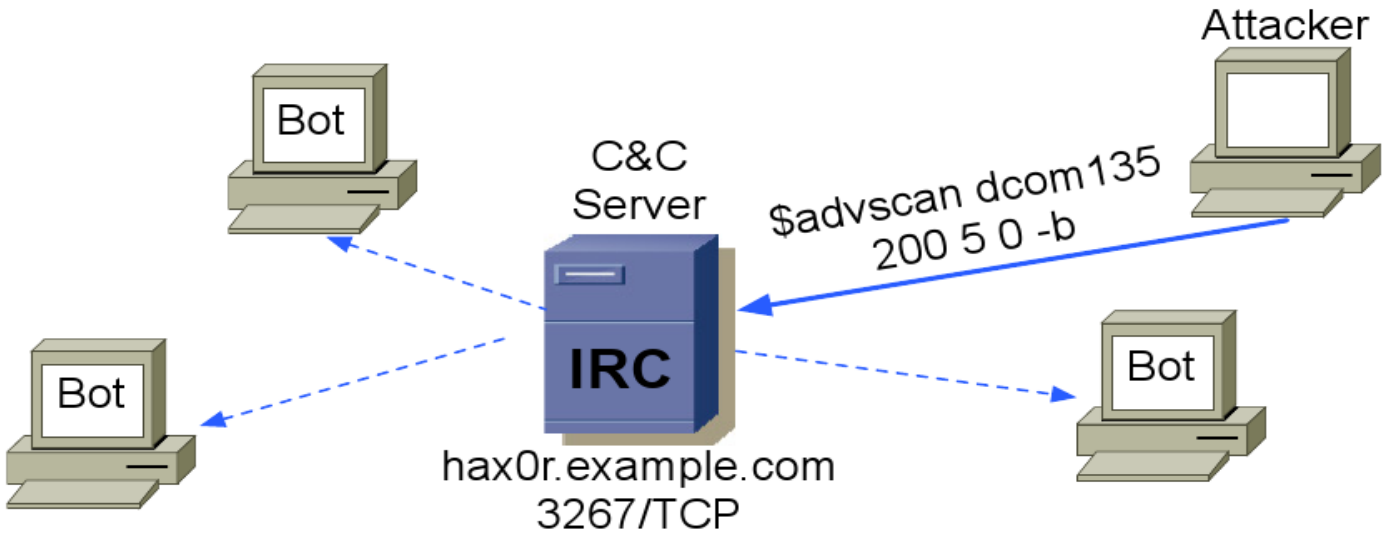
- Decentralized Command and Control mechanisms
  - P2P C&C, e.g. Storm

Centralized C&C



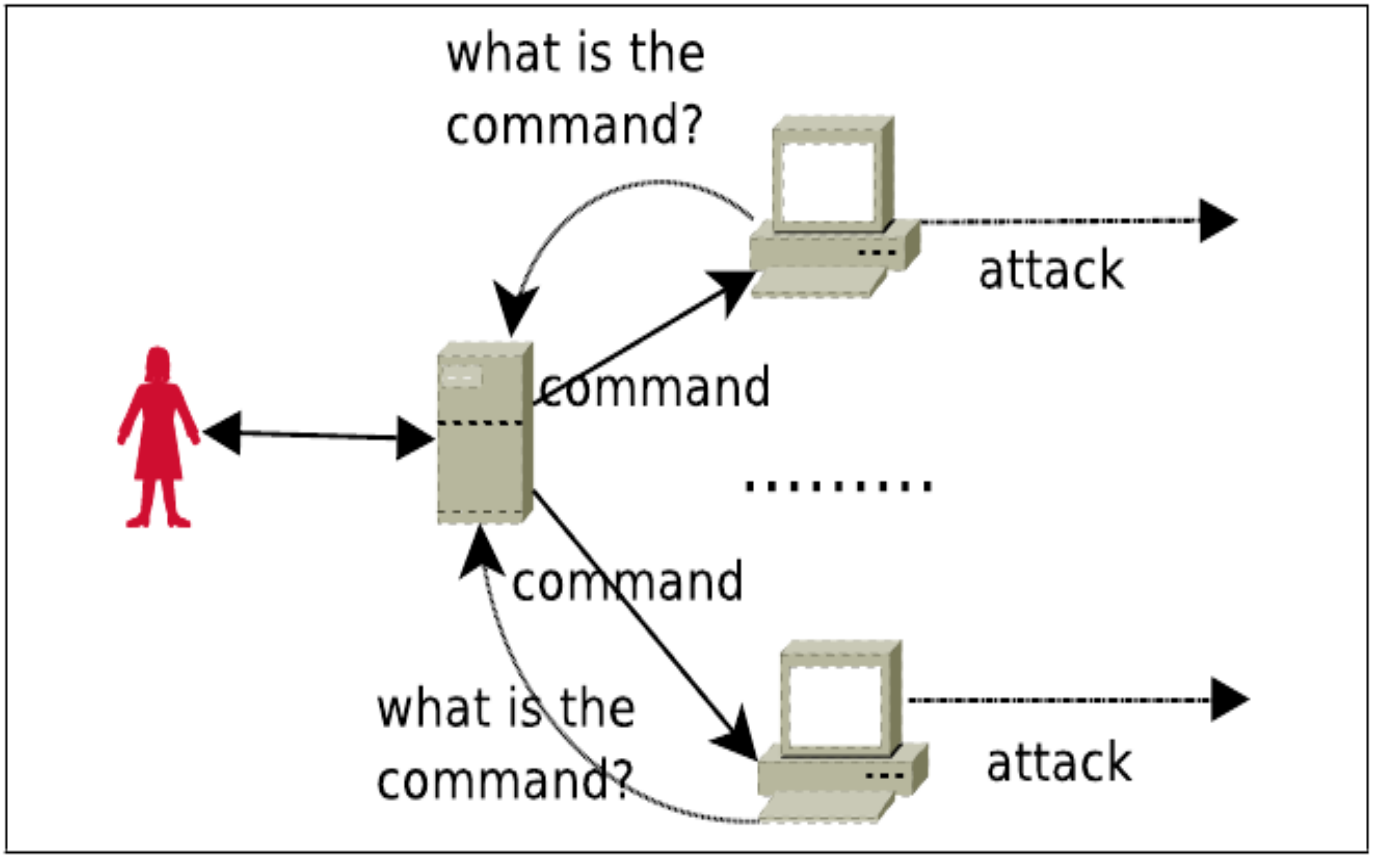
Push Style C&C using IRC

- Typical communication flow using central IRC

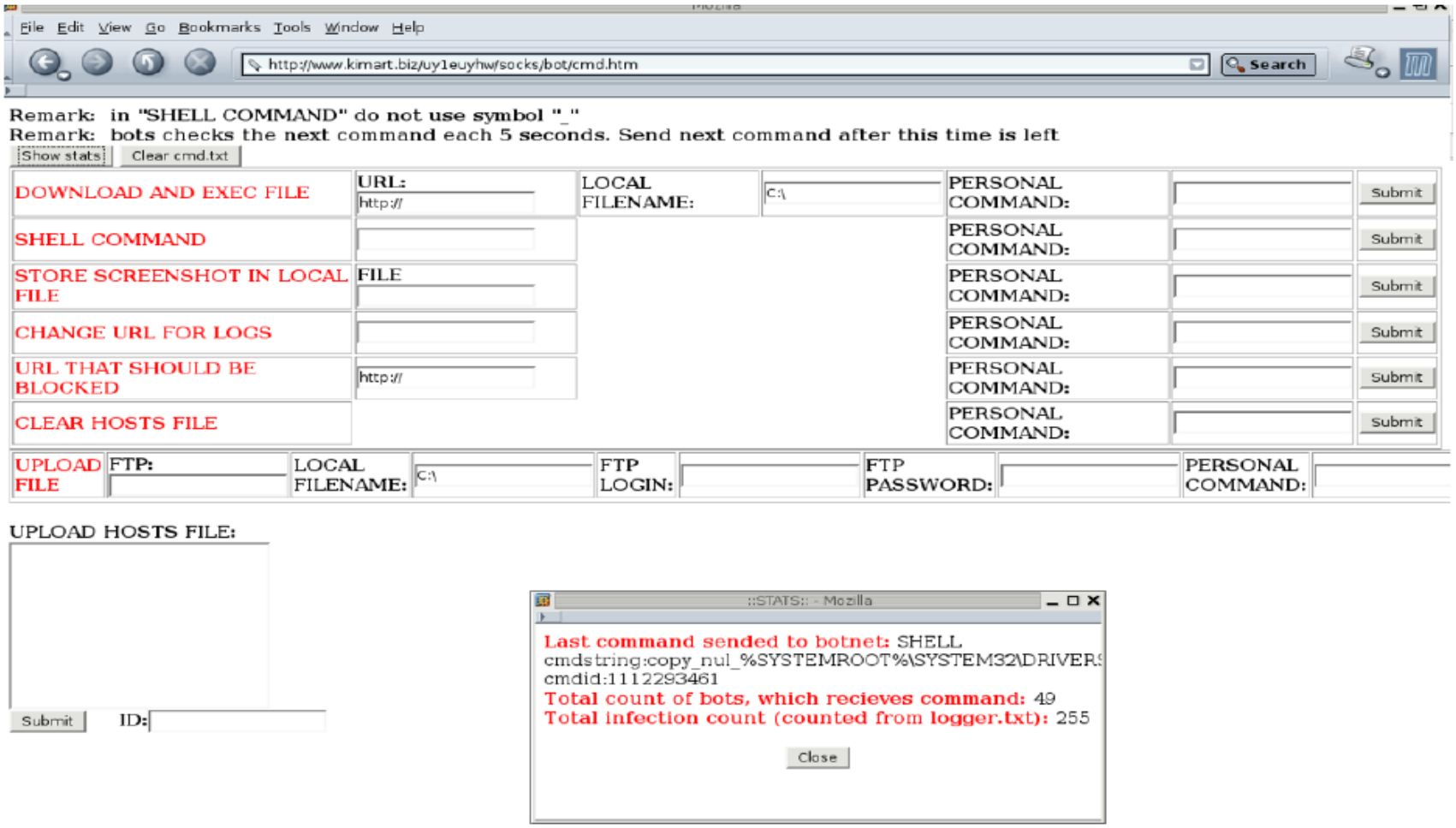


- *advscan lsass 200 5 0 -b*
- *ddos.syn XXX.XXX.XXX.XXX 80 600*

Pull Style C&C



Pull Style C&C using HTTP



Using P2P as C&C

- The best example for decentralized C&C mechanisms where the nodes of the botnet behave both as a server and a client

- Therefore
  - more difficult to catch the attacker (botmaster)
  - more robust
    - even if some of the nodes in the network are shut down, the gaps in the network are closed, and the network continues its activities

Using P2P as C&C

- Most of the well-known P2P botnets used the Overnet network which is a Kademlia-based protocol
  - Bots do not directly send information to each other, but publish a piece of information that is issued by the botmaster when she wants to perform an activity
  - Every day, to check weather a command is issued or not, the bots search for 32 different keys, which are computed with a function that takes the current date and a random number between 0 and 31 as a parameter
  - Since the attacker knows which keys are searched for everyday, she can publish the commands with the keys to be searched

Interesting Bot Commands

- At least two types of commands
  - DoS attacks (e.g. SYN- and ACK-flooding attacks)
  - Update mechanism
- Other popular commands
  - Open proxy to send spam
  - Keylogger or other identity theft

Other Bot Features

- Most bots are packed
  - UPX
  - Morphine, ASProject, Petite, tELock,...
  - Armadillo, Themida
- Anti-debugging mechanisms
  - Detection of Vmware
    - Redpill
  - Detection of debuggers

Redpill (Remember Matrix?)

```
int swallow_redpill () {
     unsigned char m[2+4], rpill[] = "\x0f\x01\x0d\x00\x00\x00\x00\xc3";
        *((unsigned*)&rpill[3]) = (unsigned)m;
        ((void(*)())&rpill)();
        return (m[5]>0xd0) ? 1 : 0;
     }
```

Checks the address of the interrupt descriptor table (IDT) with the SIDT instruction

Some Popular Botnets

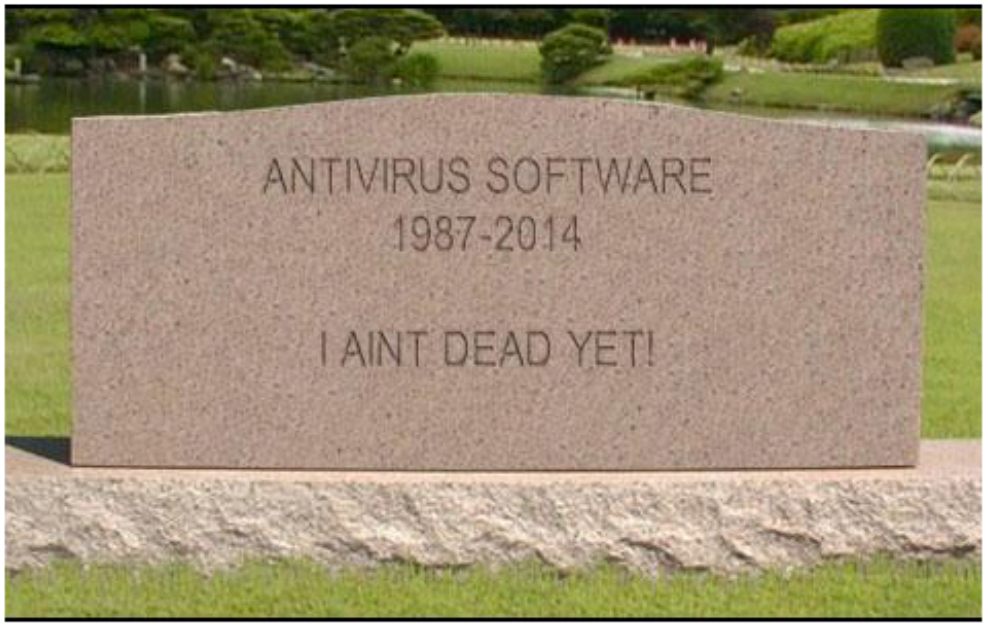
- Zeus
  - 3.6 > million compromised computers
  - Uses keylogging techniques to steal sensitive data
- Koobface
  - 2.9 > million compromised computers
  - Uses social networking sites to spread
- TidServ
  - 1.5 > million compromised computers
  - Spread through e-mail spam
  - Uses rootkit techniques to run inside common Windows services

AV te but not a st…

Symantec Says Antivirus Is Dead, World Rolls Eyes

May 07, 2014 11:55 AM EST | 32 Comments

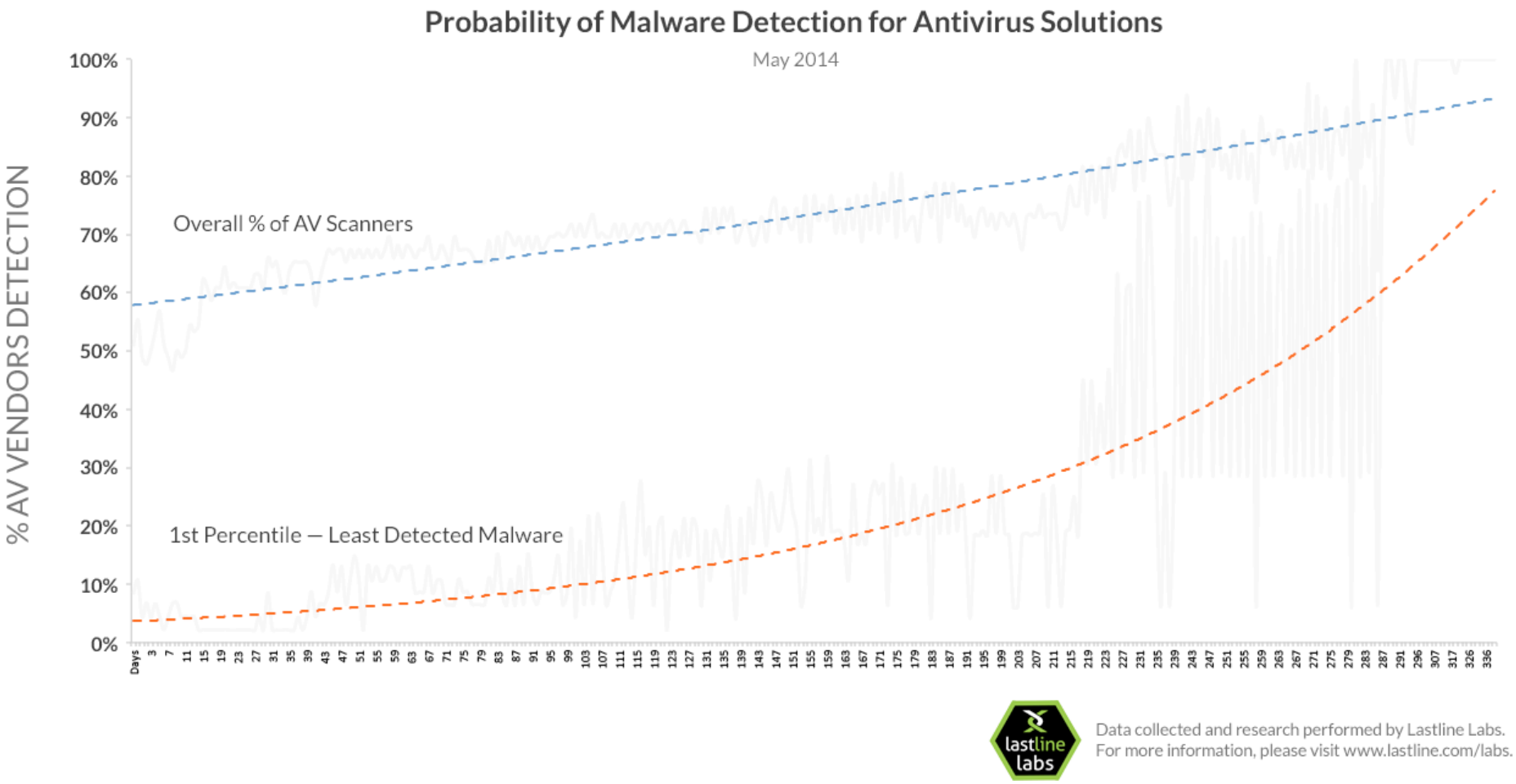
By Max Eddy



Earlier this week, Symantec's senior vice president Brian Dye declared to the **Wall Street Journal** that antivirus "is dead." That's a bit surprising, considering it still accounts for a reported 40 percent of Symantec's revenue. Plus, Symantec continues to churn out Editors' Choice award winning products like **Norton 360**. So is AV really dead? The short answer is "no," and the long answer is "no no no no no nonononononononono."

Lastline Labs: AV Can't Keep Up

Antivirus systems take months to catch up to highly evasive threats.



lastline labs

Data collected and research performed by Lastline Labs.
For more information, please visit www.lastline.com/labs.

Current State of Affairs

- Anti-virus systems are not enough
  - Malware modifies itself to evade detection
- Manual analysis of threats requires an enormous amount of resources
  - Cannot scale, reaction time in the order of days or weeks
- We need to be leading in the arms-race

How Have Security Technologies Evolved?

Emergence of Behavior-Based Detection

Key Idea

- Why not just run or open the suspicious file and see how it behaves?
- This approach is generally-known as *sandboxing*
- The sandbox typically uses a virtualized, instrumented environment
  - The system logs the behaviors of the file

Sandbox-Based Detection Is Popular

- There are many security products now
  - Sandboxing is often a component that is used for unknown files
- These sandboxes often vary in quality
  - A sandbox can be very simple, or can be more sophisticated based on its design

Evasion of Behavior-Based Detection

- Bad guys are not stupid
  - They have received the news that behavior-based detection is what everyone's using now
- Just like signature-based detection systems were evaded in the past
  - Behavioral evasions tricks have emerged

One of The First Tricks That Emerged: Red Pill (Remember Matrix?)

- A Virtual Machine (VM) is often used to run the code during analysis and detection
- The red pill test allows you to find out if you're running in a VM
- There are *many* ways of launching evasions like that

Some Dynamic Evasion Tricks

- Checking for specific artifacts in the virtualized OS
- Checks on CPU features that indicate VM
- Looking for running processes and imitating them
- Waiting for someone to click on something
- Delaying the execution until analysis system gives up

An Emerging Trick: Stalling Loops

- Simple piece of code that takes milliseconds to execute on your laptop, but hours to run in a virtualized detection system


```
1 unsigned count, tick;
2
3 void helper() {
4   tick = GetTickCount();
5   tick++;
6   tick++;
7   tick = GetTickCount();
8 }
9
10 void delay() {
11   count=0x1;
12   do {
13     helper();
14     count++;
15   } while (count!=0xe4e1c1);
16 }
```

Targeted, Government-Sponsored Attacks

Stuxnet: Who did it?

National Security

Stuxnet was work of U.S. and Israeli experts, officials say

A 389

By Ellen Nakashima and Joby Warrick June 2, 2012

Follow @nakashimae Follow @jobywarrick

A damaging cyberattack against Iran's nuclear program was the work of U.S. and Israeli experts and proceeded under the secret orders of President Obama, who was eager to slow that nation's apparent progress toward building an atomic bomb without launching a traditional military attack, say current and former U.S. officials.

The origins of the cyberweapon, which outside analysts dubbed Stuxnet after it was inadvertently discovered in 2010, have long been debated, with most experts concluding that the United States and Israel probably collaborated on the effort. The current and former U.S. officials confirmed that long-standing suspicion Friday, after a New York Times report on the program.

The officials, speaking on the condition of anonymity to describe the classified effort code-named Olympic Games, said it was first developed during the George W. Bush administration and was geared toward damaging Iran's nuclear capability gradually while sowing confusion among Iranian scientists about the cause of mishaps at a nuclear plant.

Most Read

1 Couple seemed quiet and withdrawn — until explosion of violence 

2 'I've never witnessed something so sad in my life': Stories of the Calif. shooting victims 

3 Islamic State money-making streams take a hit as it loses territory 

4 U.S. eliminates a mid- to high-level ISIS figure every 2 days, official says 

5 UK: Maoist cult leader guilty of rape, sex abuse charges 

The Most Popular All Over

The Hill
UnitedHealth CEO regrets entering ObamaCare 

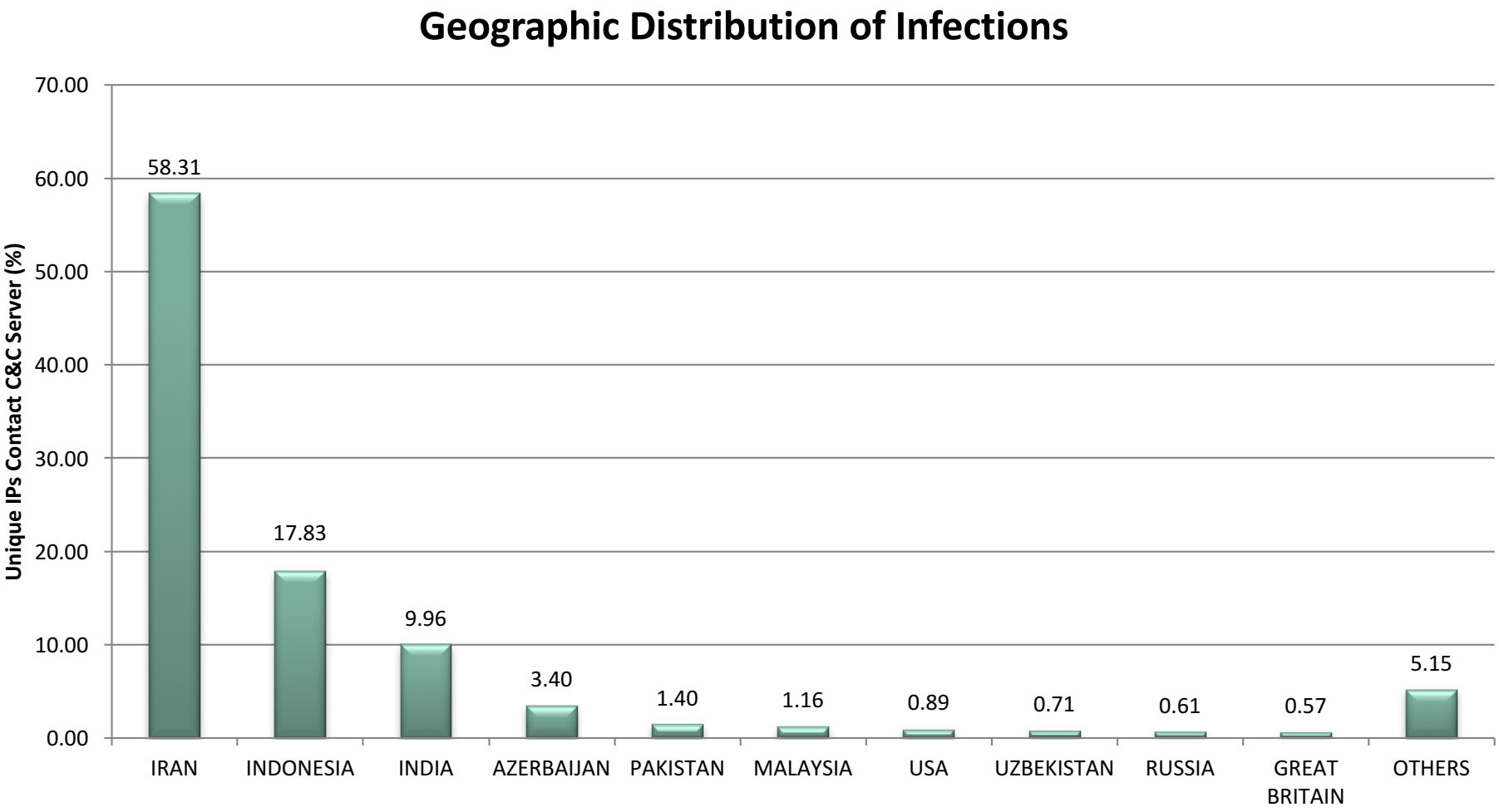
Agenda

1. Stuxnet Capabilities
2. Network Distribution Tactics
3. Intel & Targets
4. Sophistication & Success
5. Solutions & Lessons Learned

Stuxnet Features

- Discovery disclosed in July, 2010
- Attacks industrial control systems likely an Iranian uranium enrichment facility
- Modifies and hides code on Siemens PLCs connected to frequency converters
- Contains 7 methods to propagate, 4 zero day exploits, 1 known exploit, 3 rootkits, 2 unauthorized certificates, 2 Siemens security issues, 1 target.
- 3 versions, June 2009, March 2010, April 2010

Stuxnet is targeted: Iran



PLCs: Programmable Logic Controller

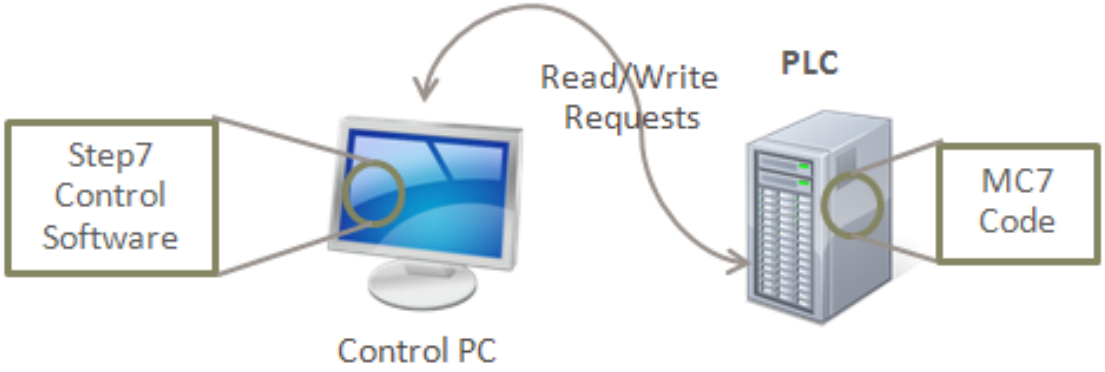
- Monitors Input and Output lines
  - Sensors on input
  - switches/equipment on outputs
  - Many different vendors
- Stuxnet seeks specific Models
  - s7-300 s7-400

Stuxnet is Targeted

Targeting a **Specific type of PLC**

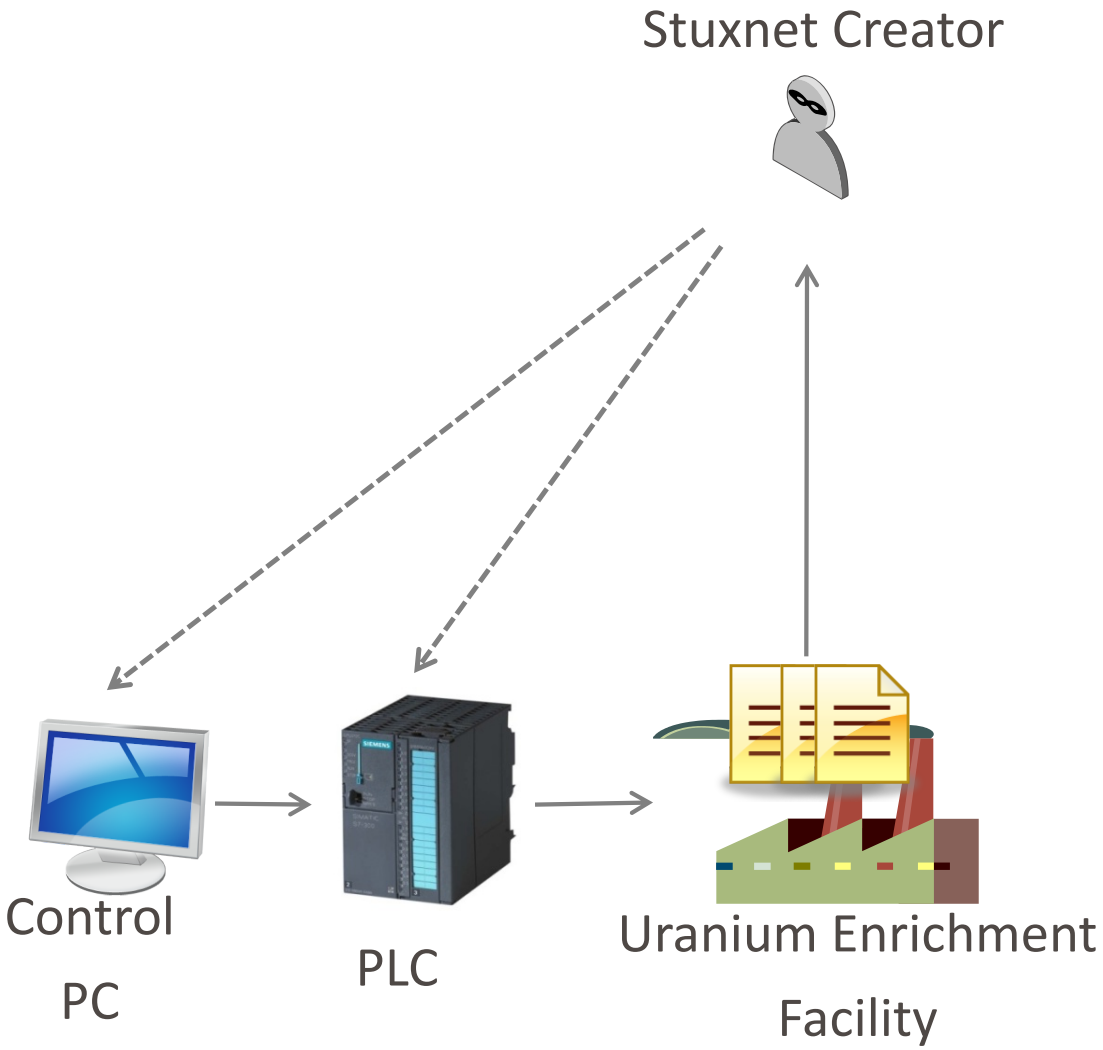
Searches for a **Specific Configuration**

Programming a PLC

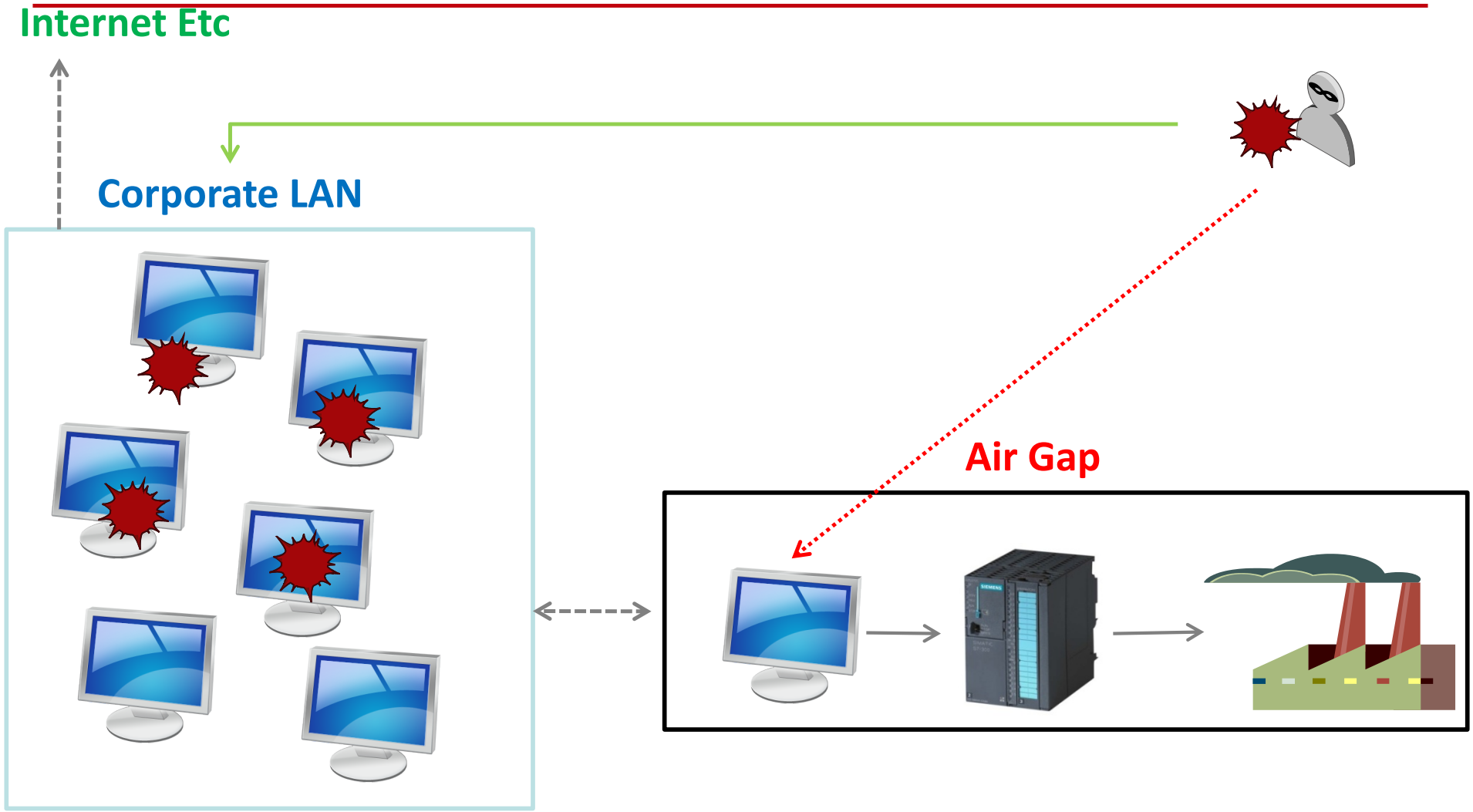


- Simatic or Step 7 software
  - Used to write code in STL or other languages
- STL code is compiled to MC7 byte code
- MC7 byte code is transferred to the PLC
- Control PC can now be disconnected

Attack Preparation



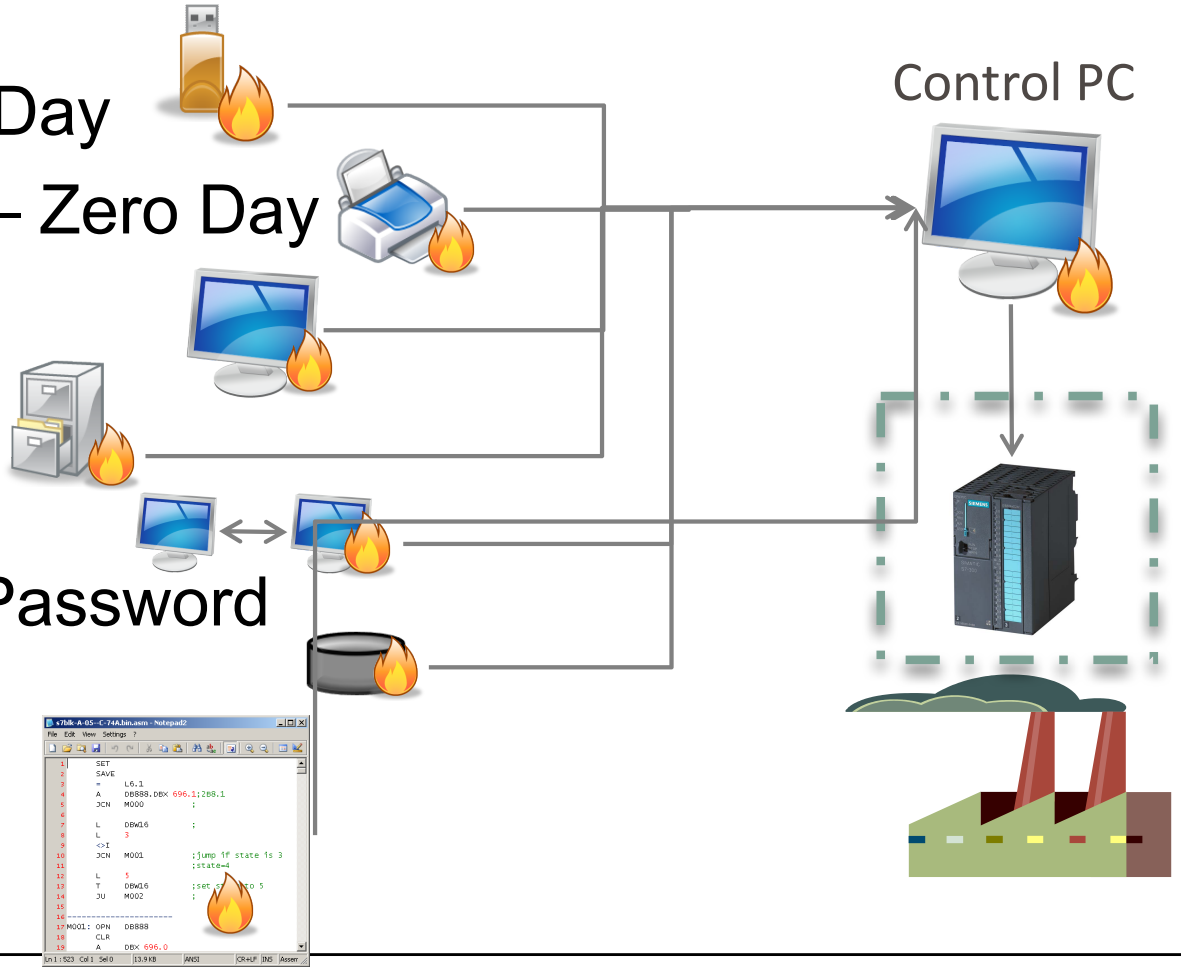
Attack Considerations



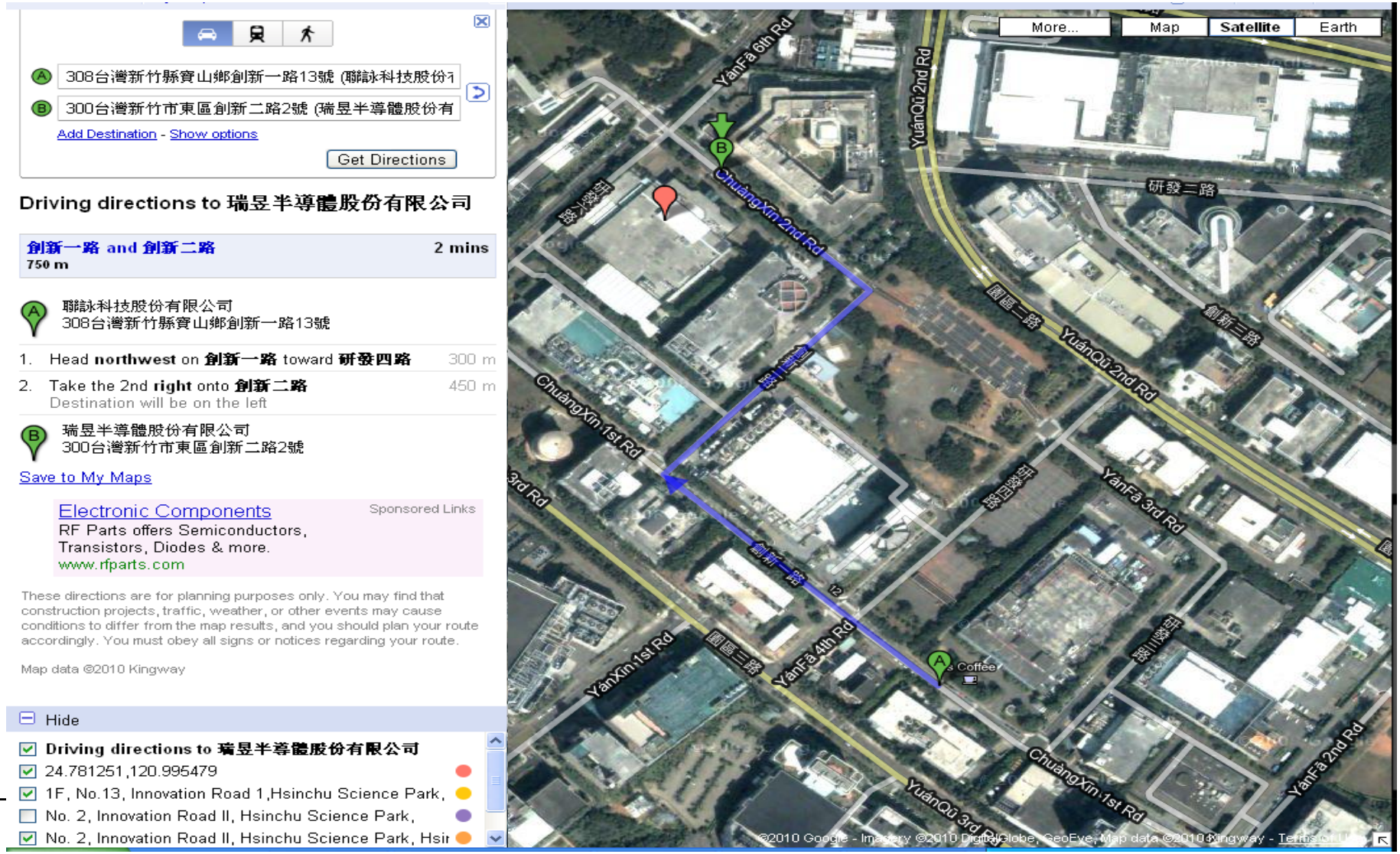
How Stuxnet Attacks Corporations

Stuxnet uses 7 different methods to propagate!

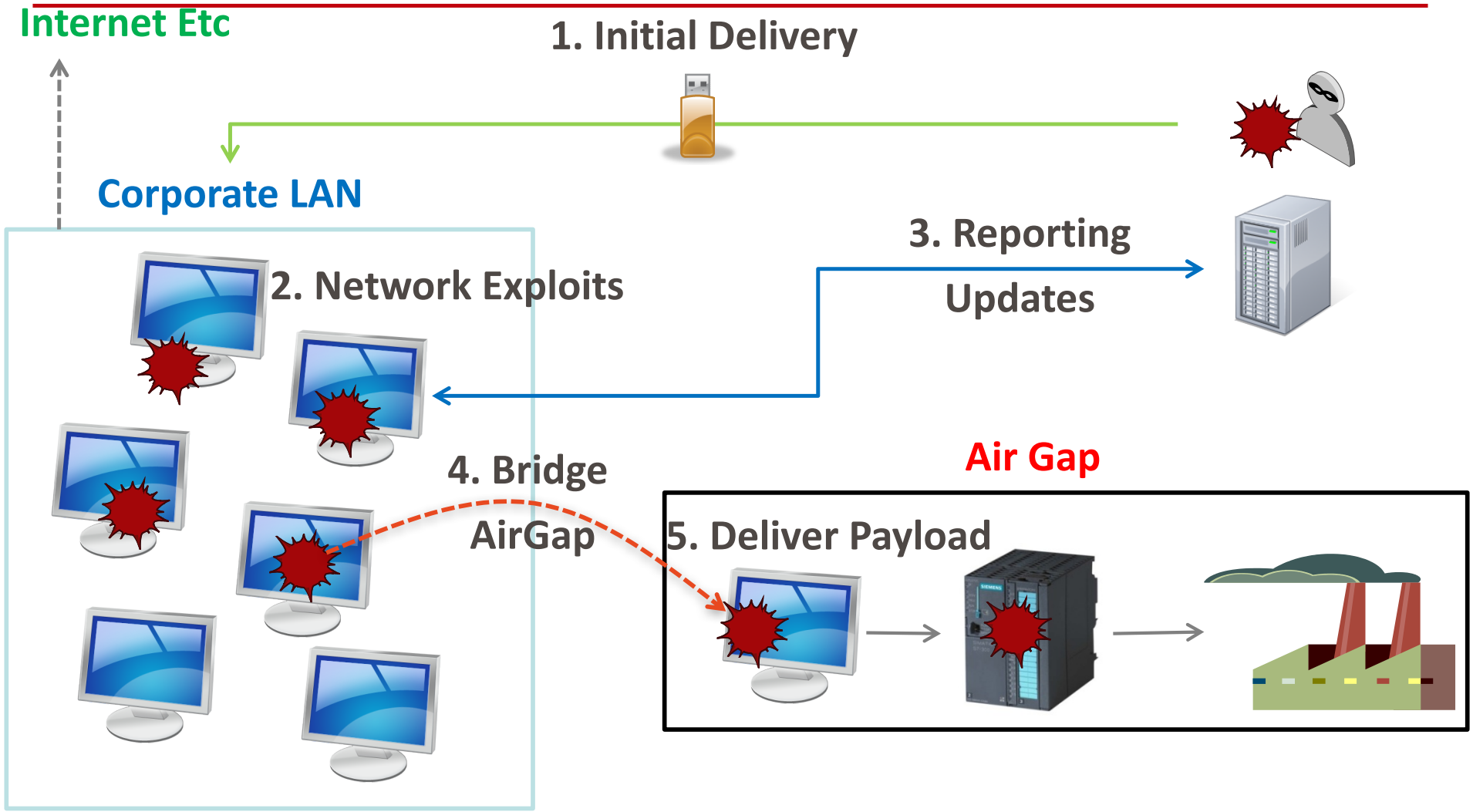
1. USB drives – Zero Day
2. Print Spooler Vuln – Zero Day
3. Ms08-067 Vuln
4. Network Shares
5. P2P sharing
6. Wincc Hardcoded Password
7. Step7 projects



Stuxnet Windows Rootkit



Attack Execution



Delivering the threat

- Stuxnet targeted specific companies in Iran
- Only 10 initial targets
- Resulting in over 14k infections
- Research was needed to identify valuable targets
- Companies connected to Uranium enrichment
- Hope to infect someone who would visit a Uranium enrichment facility
- Someone who worked on Uranium enrichment projects
- Actual delivery method is unknown

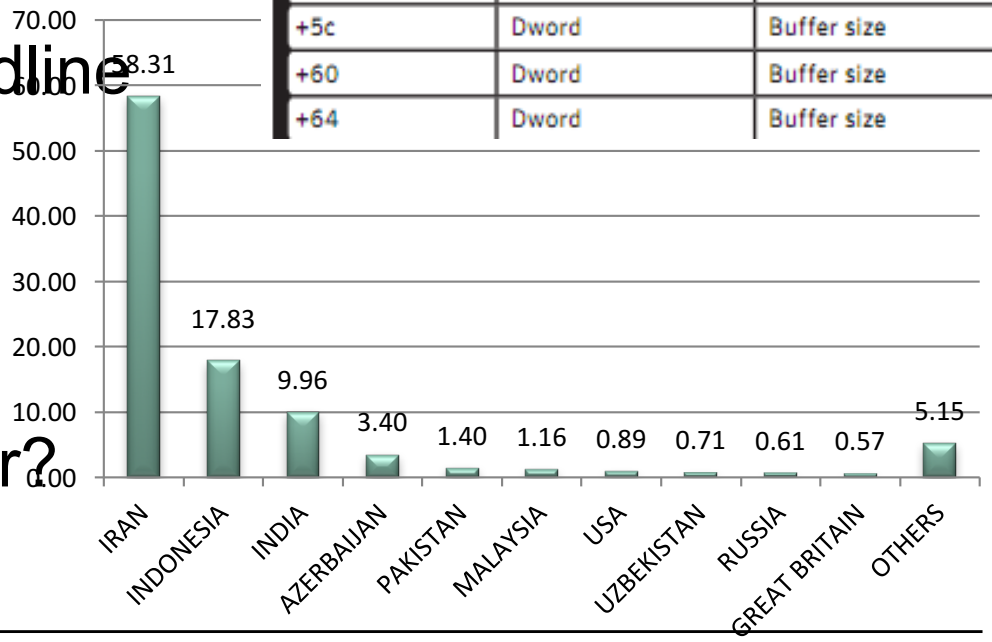
Limited Spr

- Attackers wanted limited spread
- No Internet capable exploits use
- USB exploit only infects 3 mach
- USB exploit has deadline of 21
- All exploits have a deadline

- Large configuration file
- ~430 different settings
- Why did it spread so far?

Table 10
Configuration Data

Offset	Type	Description
+0	Dword	Magic
+4	Dword	Header size
+8	Dword	Validation value
+C	Dword	Block size
+10	Dword	Sequence number
+20	Dword	Performance Info
+24	Dword	Pointer to Global Config Data
+30	Dword	Milliseconds to Wait
+34	Dword	Flag
+40	Dword	Pointer to Global Config Data
+44	Dword	Pointer to Global Config Data
+48	Dword	Pointer to Global Config Data
+58	Dword	Buffer size
+5c	Dword	Buffer size
+60	Dword	Buffer size
+64	Dword	Buffer size



Why did it spread so far?

- Zero .lnk vulnerability wildly successful
- Step7 project infection very successful
- Misunderstanding of how contractors interact
- Misunderstanding of how connected companies are communicating
- Intended?
- Needed to be more aggressive to succeed?

Was Stuxnet Successful

- We don't know.

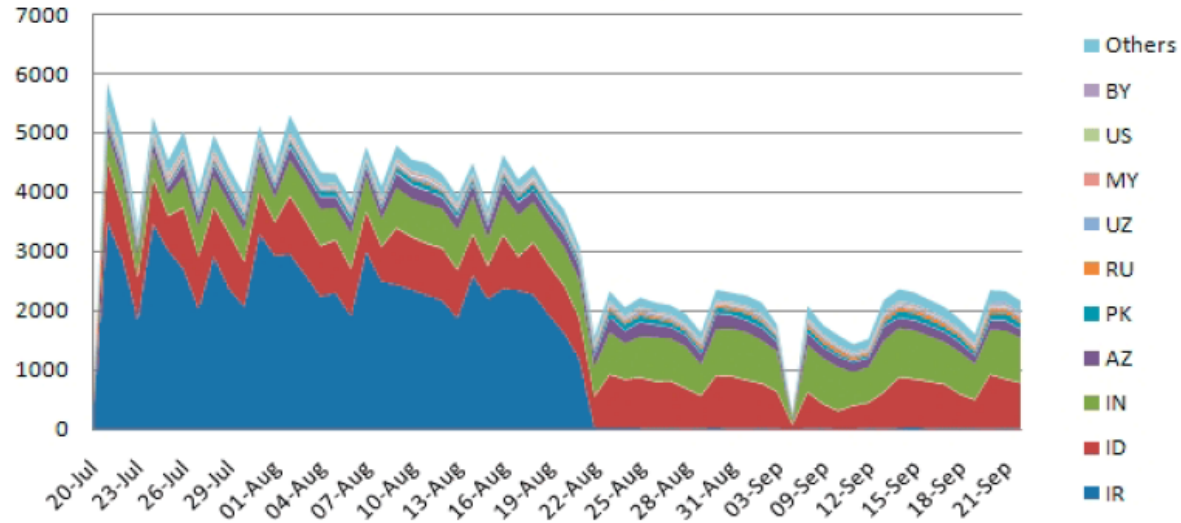
- 1 year in the wild undiscovered
- Over 100k infections
- Majority in Iran
- Natanz shut down
- Industrial Companies Infected
- Reports of infections at Natanz and Busheir
- IAEA report states 1000 centrifuges offline in Nov 2009

Was Stuxnet Successful

- We don't know.

- Discovered 3 months after USB zero day added
- No report of centrifuges out of action after March 2010
- Gained high media a
- Analysis performed
- Iranian authorities a

Figure 5
Rate of Stuxnet infection of new IPs by Country



Sophistication

- First threat to target hardware
- Targets Uranium Enrichment
- Large amount of code
- Very configurable
- 4 zero days
- Long Reconnaissance phase
- Needed Hardware for testing
- Targets 95/98,Win2k,Winxp,Vista,Win7…
- 3 Rootkits
- PLC programming knowledge

Sophistication

- It was discovered
- No advanced encryption
- C&C infrastructure easily taken down
- Infection information stored
- Blue screens?? (unconfirmed)
- P2P not protected
- Escaped outside of Iran

New Version

- Not simple to create new version
- Cannot just drop in new zero days
- Target specific information required
- PLC programming knowledge
- Exploit knowledge

- Real danger is the idea
- Now people know it can be done
- People can start their own projects knowing it is possible

Solutions & lessons learned

- Insider threat is significant – Employees are major risk
- IP is extremely valuable, protect it at all costs
- Monitor systems and networks
- Watch for red flags
- Implemented real air gaps
- Or accept this is not possible and protect computers inside the air gap more vigorously
- White listing, behavior blocking and reputation based solutions can mitigate threat.
- Device blocking – USBs, contractor laptops, etc..
- Vigilance is key