Lecture #5: Network Security & Monitoring #1

WPI CS4516 Spring 2019 D term

Instructor: Lorenzo De Carli (ldecarli@wpi.edu)

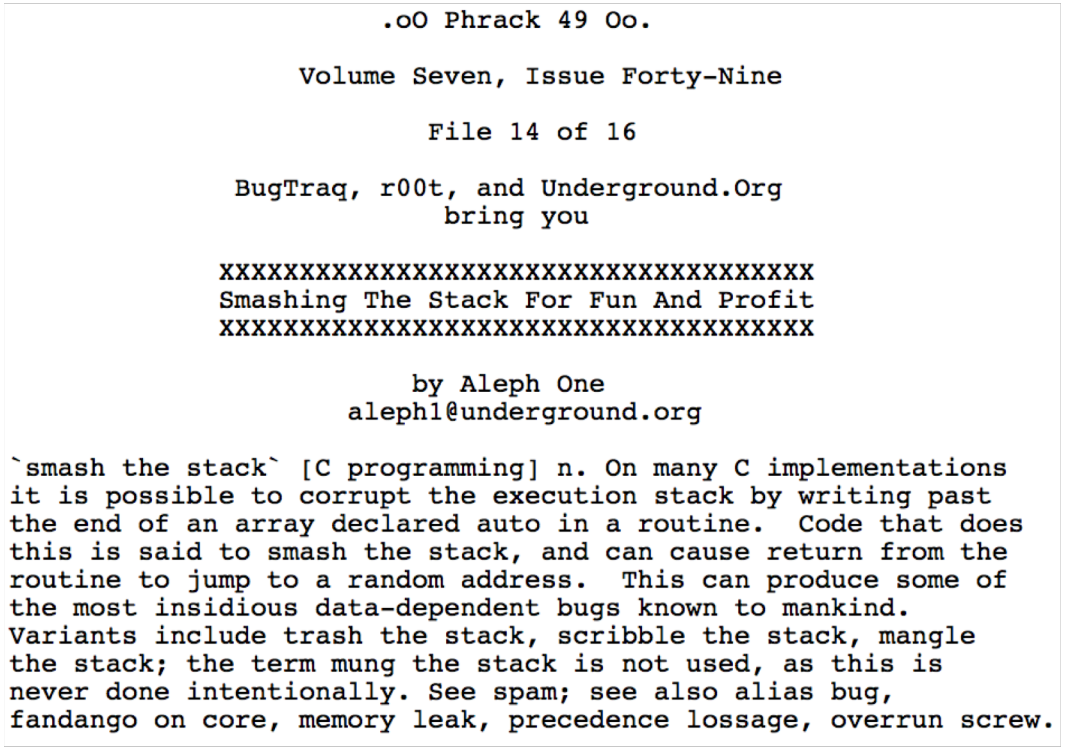
(slides include material from Christos Papadopoulos, CSU)

Attacks

- Impact
  - Denial or degradation of service
  - Compromised information
  - Economic implications
- Our focus
  - Classify various attacks
  - Begin discussion of countermeasures

Earlier Attacks (early ‘90s)

- **System intrusions**
  - Cracking passwords and obtaining access to systems
  - Exploiting vulnerabilities (buffer overruns) to gain access to systems
  - See tools such as Metasploit for examples



.oO Phrack 49 Oo.

Volume Seven, Issue Forty-Nine

File 14 of 16

BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

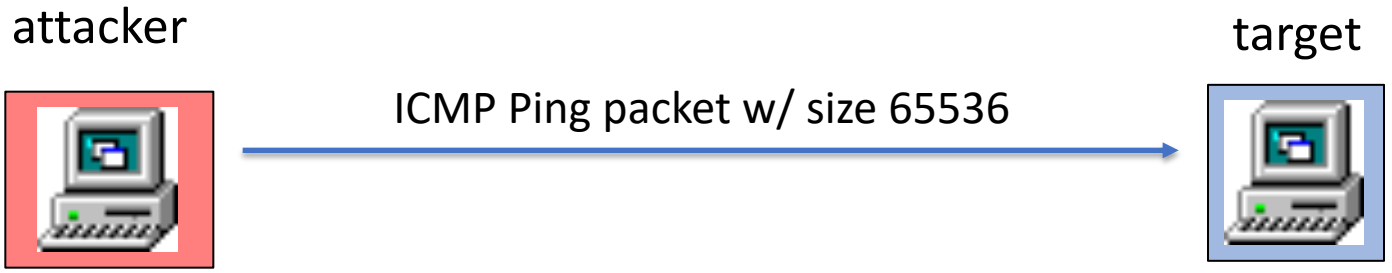
by Aleph One
aleph1@underground.org

`smash the stack` [C programming] n. On many C implementations it is possible to corrupt the execution stack by writing past the end of an array declared auto in a routine. Code that does this is said to smash the stack, and can cause return from the routine to jump to a random address. This can produce some of the most insidious data-dependent bugs known to mankind. Variants include trash the stack, scribble the stack, mangle the stack; the term mung the stack is not used, as this is never done intentionally. See spam; see also alias bug, fandango on core, memory leak, precedence lossage, overrun screw.

Denial of service (late ‘90s)

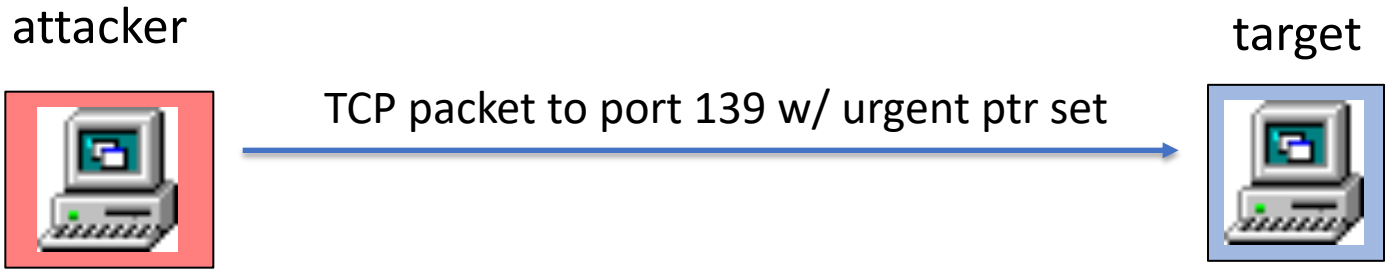
- Later attacks: **Distributed Denial of Service (DDoS)**
  - What is it?
  - **Flood a machine** (e.g. a website) with traffic so as to **make it temporarily unavailable**
  - Can do this by using **intrusion techniques**, or **exploit vulnerabilities** (e.g. Windows IP fragmentation) to crash machine…
  - …or just by **sending a lot of traffic** to it!

Ping of death



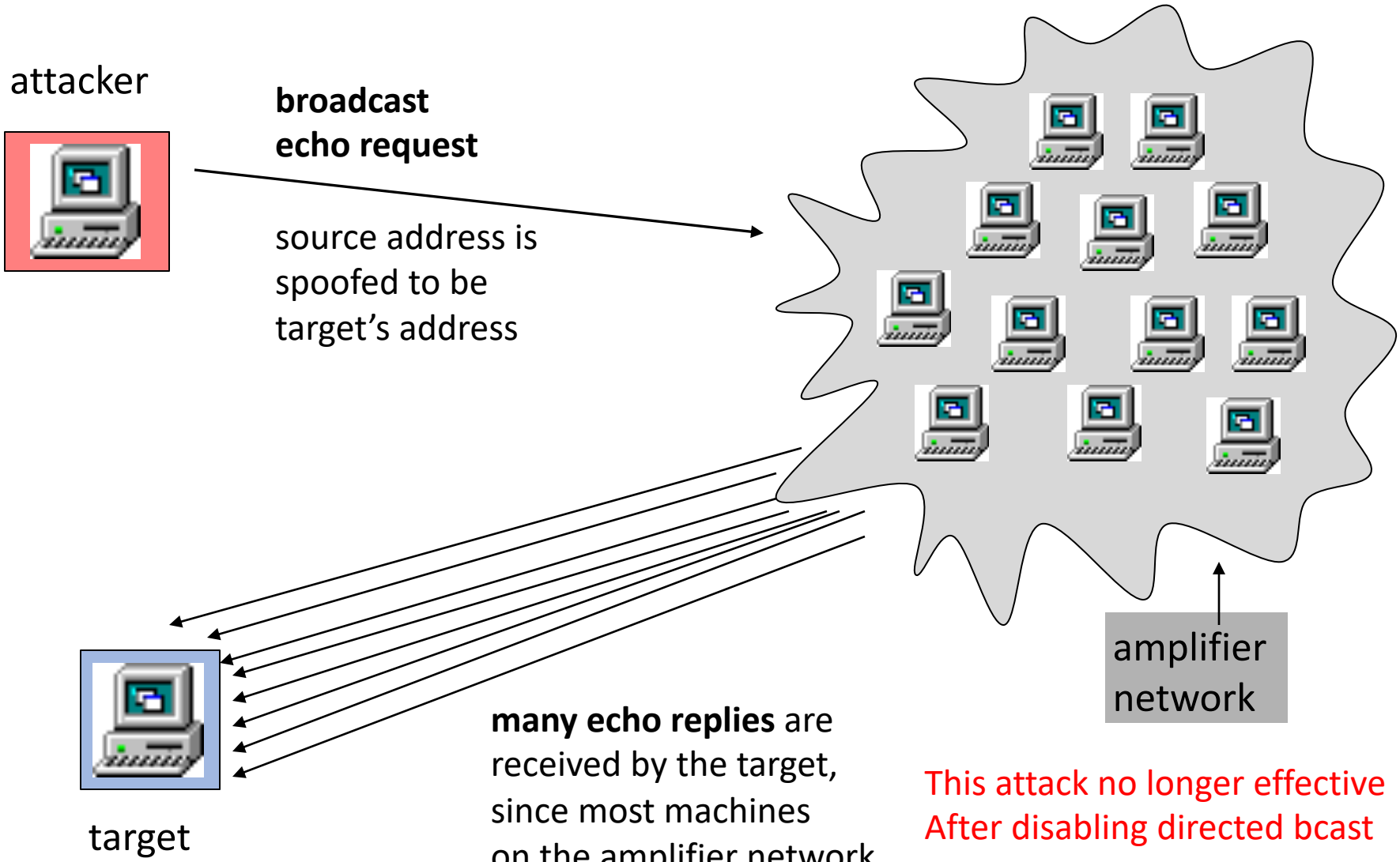
- Attacker sends large ICMP ping packet
- Packet gets fragmented
- Target reconstructs it; because reconstruction buffer is not designed with large packet in mind, **buffer overflow** is triggered

WinNuke



- Attacker send malicious (but correct!) TCP packet w/ urgent ptr set
- Windows 95/NT can’t process the packet correctly and **crashes**

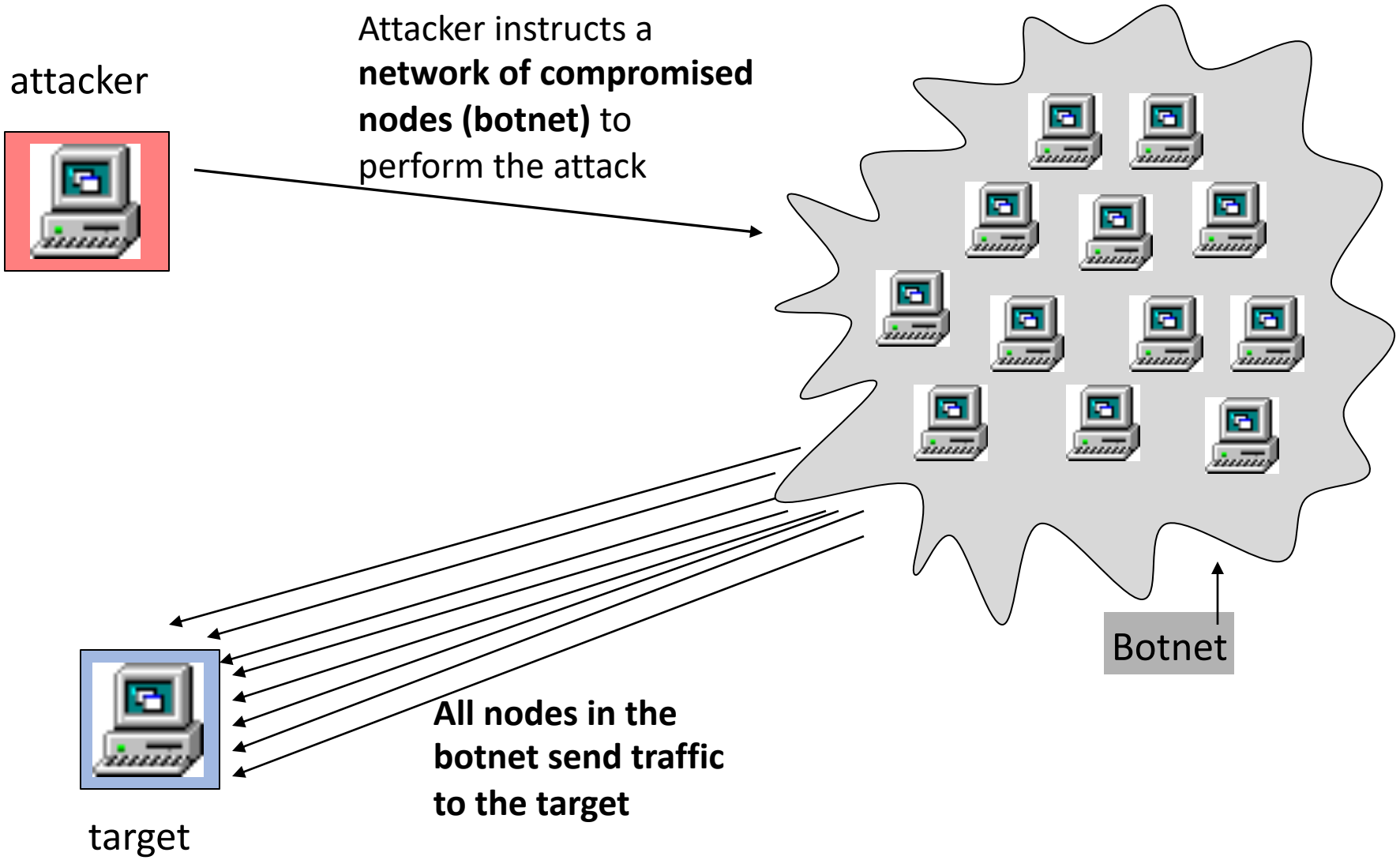
Smurf Attack



Distributed Denial-of-Service

- Two key ideas
  - Compromise a **large number of systems** over time
  - Use them to ***simultaneously*** **generate large volume of traffic**
    - Use source-spoofing for evasion
- Traffic generator examples
  - ICMP or UDP flood: can overwhelm routers that forward packets
  - TCP SYN flood: can overwhelm servers by causing them to maintain state
- Much of this can be **automated**
  - Master-slave architecture

Botnet-based denial of service

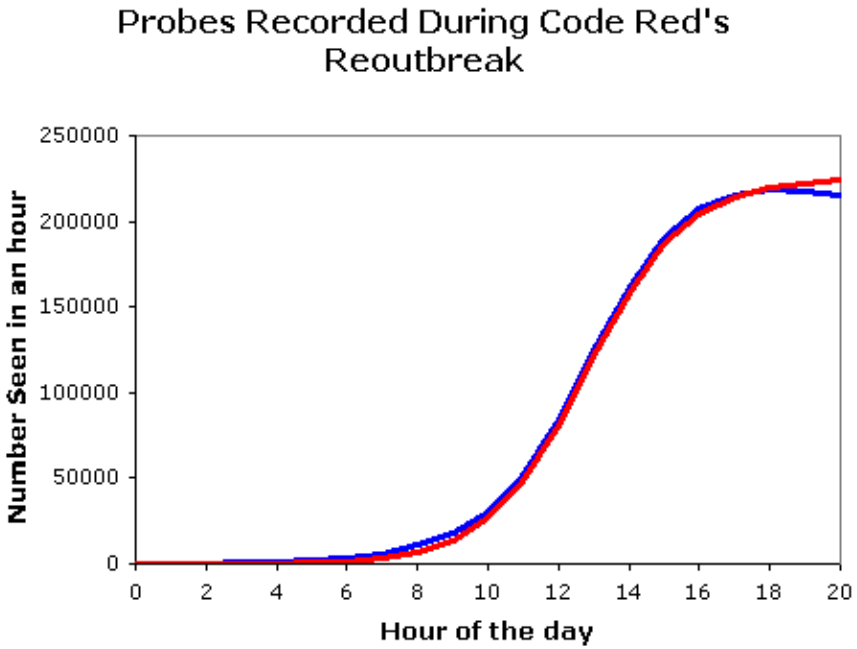


Third-Generation Attacks (early 2000's)

- Use **self-propagating code techniques** (**worms**) to intrude machines
- Key ideas:
  - **Exploit *one* vulnerability on a large number of deployed systems** to **rapidly propagate code** that gains access to a system
  - *(viruses* are slightly different from worms and require human intervention such as reading email to spread)
- Goals vary
  - Launch DDoS, although the very act of self-propagation may overwhelm networks with traffic
  - Extortion (e.g., gambling sites)
  - Use machines for covert actions (e.g., SPAM, scanning, etc.)

Anatomy of a Worm

- Initiated from a **single machine** or using a **hit list**
- Scans other machines for a vulnerability
  - brute-force (e.g. randomly scanning IP addresses
- **Uploads code** to infected machine
- **Repeats** process



From "The Spread of the Sapphire/Slammer Worm", Moore et al. 2003

Worm Examples

- Code Red I, July 2001
  - Uses Microsoft web server vulnerability
  - Randomly generates IP addresses, intrudes machines
  - Set to attack whitehouse.gov
- Code Red II
  - Uses same vulnerability
  - Slightly more sophisticated address probing technique
- Nimda
  - Scanning technique different from Code Red II
- Sapphire/Slammer, Jan 2003
  - Uses small UDP packets, so very fast
  - By contrast, previous worms had larger scanners and used TCP

Another problem: attack attribution

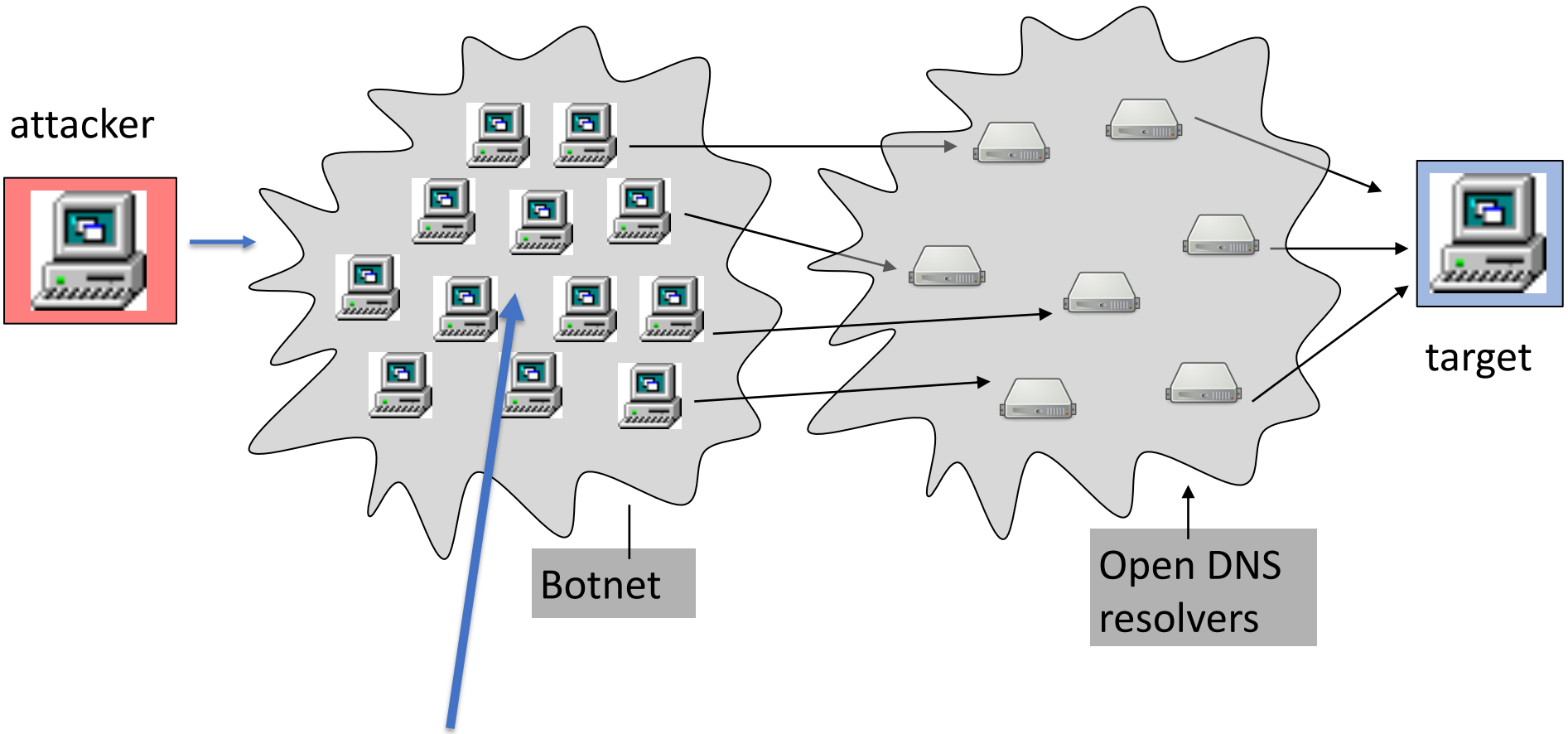
- **Does it matter?** For a while people thought so
- Source IP may be spoofed - so how can we trace spoofed IP packets to their actual source?

Attacks today

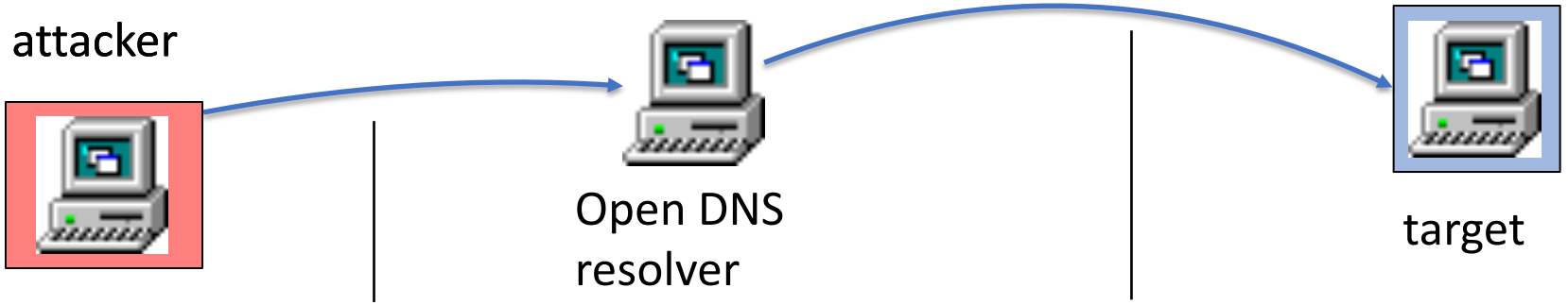
DDoS today

- Some work through **amplification**
- **Attribution** not so important
- We'll see why in a few slides

DNS amplification DDoS

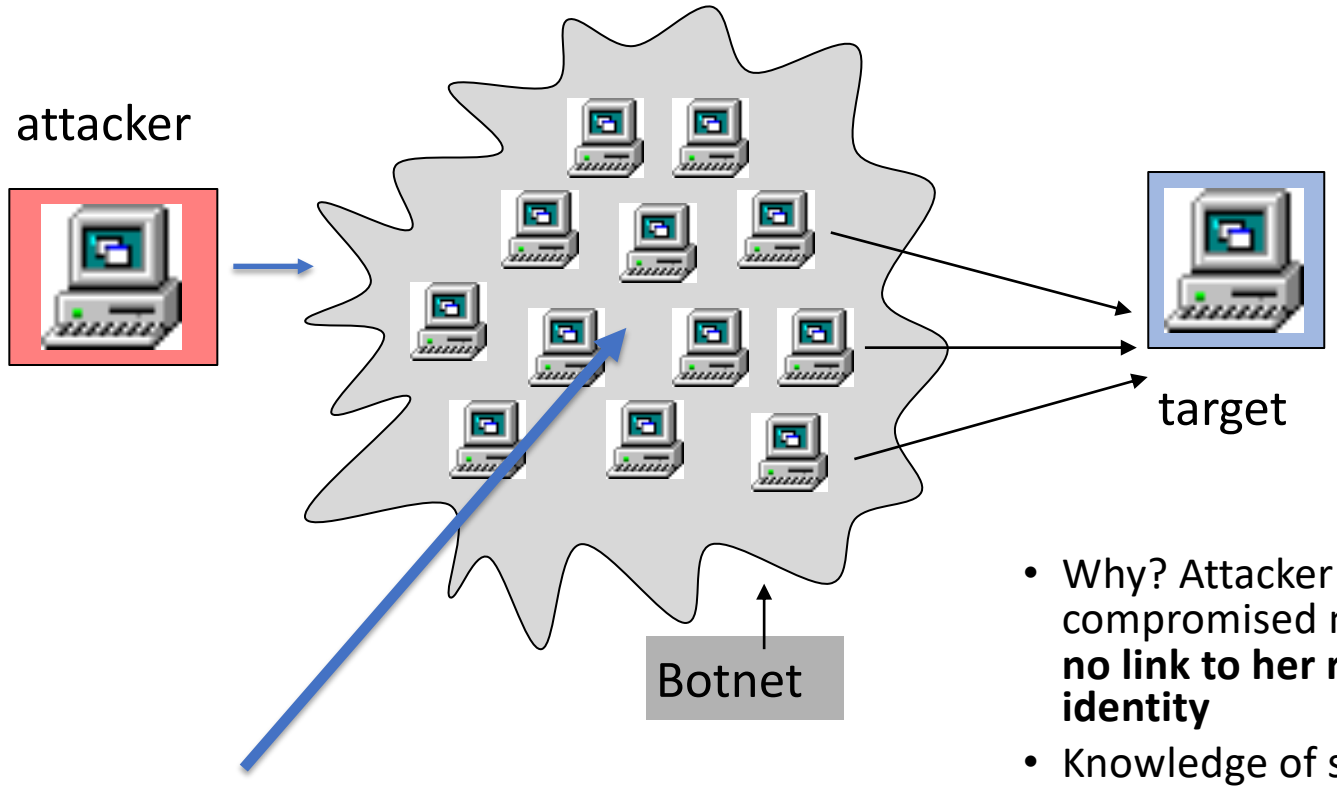


Inside a DNS amplification attack



Other amplification attacks are possible (e.g. using NTP) - really, any protocol that **replies to small requests with large replies** and **does not perform user authentication/filtering** is suitable

Attribution is moot



In a direct attack, **no attempt to mask nodes' source addresses**

- Why? Attacker uses compromised nodes **w/ no link to her real identity**
- Knowledge of source IP addresses is **not enough to stop the attack**

New DDoS frontier: smart devices

- IoT devices - smart TVs, cameras, refrigerators, etc. :-)
- Typically **not** designed with security in mind
- Easy to **compromise** and **repurpose as bots**
- An attacker with the right knowledge can easily **compromise thousands of devices in a short time**!
- Due to **massive scale**, **no need for amplification**

Scale of modern DDoS

DNS attack against krebsonsecurity.com - September 20, 2016
No amplification, just lots of compromised “smart” devices (Mirai botnet)

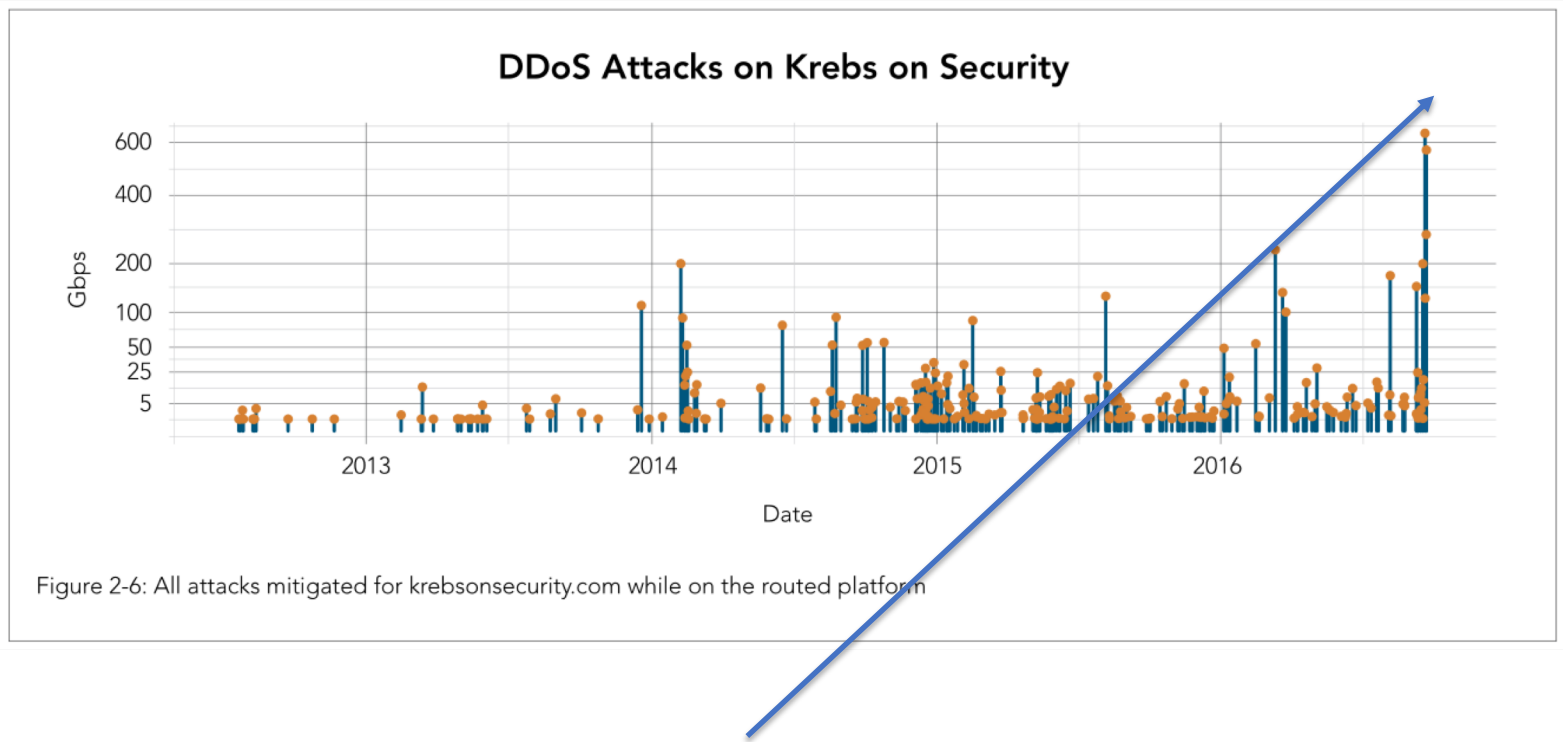


Figure 2-6: All attacks mitigated for krebsonsecurity.com while on the routed platform

665 Gbps (contrast with first highly publicized DDoS attack, against Yahoo in 2000: 1Gbps)

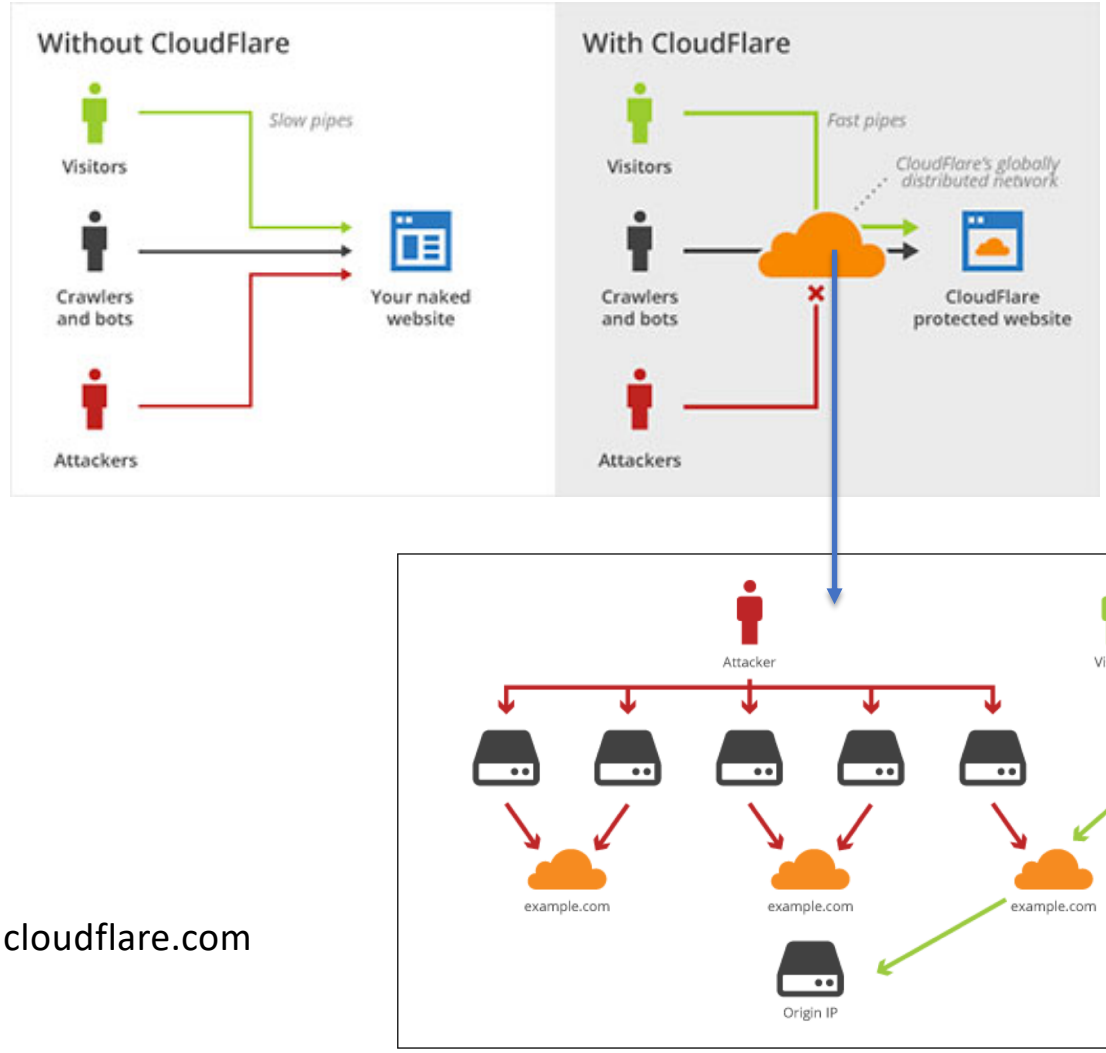
What does this mean for attribution?

- Attackers **don't try to hide source IPs**
- But does it matter when dealing with 1000's of devices all over the globe?
- Attribution may be useful to characterize and predict future attacks…
- …but **not so useful to stop them**!

DDoS mitigation

- Large Internet companies such as Cloudflare, Google, Akamai etc. offer **DDoS mitigation**
- **Reverse proxying:** traffic towards a website get distributed to a variety of network locations
- Each location applies a number of techniques to **filter out likely attack traffic**
- Only legitimate traffic is redirected to the actual site

DDoS mitigation - II



Source: cloudflare.com

DDoS mitigation - III

- How well does it work? Very well if you can afford it
  - Akamai on why they stopped giving free protection to krebsonsecurity.com: *"In an interview with The Boston Globe, Akamai executives said the attack — if sustained — likely would have cost the company millions of dollars. In the hours and days following my site going offline, I spoke with multiple DDoS mitigation firms. One offered to host KrebsOnSecurity for two weeks at no charge, but after that they said the same kind of protection I had under Akamai would cost between $150,000 and $200,000 per year."*
- Also, some DDoS mitigation companies have been known to shield malicious actors (spammers, etc.)
- On the other hand, some DDoS mitigation companies offer free mitigation for websites that may be target of politically-motivated attacks etc.

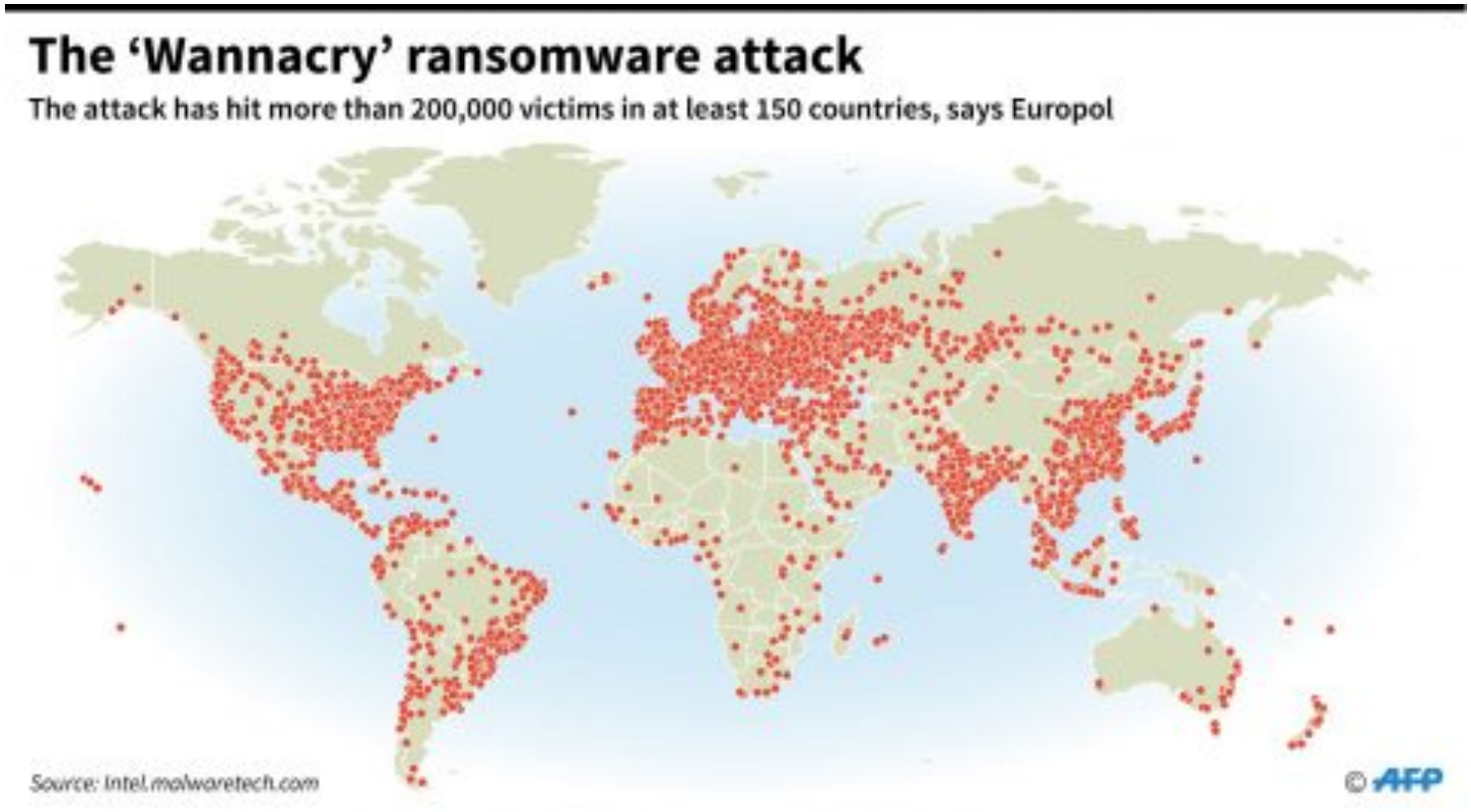
Worms today

- Now called "**malware**" :-)
- Malware focuses **not so much on active propagation**, more on **financial gain**
  - Stealing user credentials, social network accounts, banking information
  - **Large criminal economy**
  - Malware installed via **phishing campaigns** rather than through exploits
- **Massive Internet-scale disruption not a goal**
- **Can you think why?**
  - Better to "lie low" to avoid attracting the attention of law enforcement
- Botnets can still grow to thousands to millions of nodes
  - …but **propagation and activity events tend to happen at low rate**

Worms today - II

- **Rare worm outbreaks still happen**
- E.g. WannaCry and NotPetya
  - Worm encrypts files on victim machines, rendering them useless
  - Spreads using a set of well-known exploits, e.g. Windows SMB bug
- Motivation?
  - Unclear, possibly acts against specific companies which got out of hand

Worms today - III



Review of network attacks

- Break-in attempts (exploitation of vulnerability in protocol implementations or network applications)
- Malware

Prevention: analysis of traffic towards individual hosts/networks (AKA network monitoring/intrusion detection)

- Distributed denial of service

Prevention: Internet-scale solutions: ad-hoc infrastructure (content replication/distribution)

Network monitoring

- The techniques we are going to review today are effective against **intrusion attempts towards individual networks or hosts**
- They are generally based on **filtering and analyzing network flows** looking for signs of **malicious behavior**

Intrusion detection/prevention

- Network monitoring for the purpose of detecting attacks/malicious behavior is referred to as:
  - **Intrusion detection**, if the approach involves only **passive monitoring**
  - **Intrusion prevention**, if the system can perform **active behavior** (e.g. terminate connections)
- **Attacks of interest:** malware infections, exploits, data exfiltration, port scans, …

Attack examples/1

Code Red worm attack URL: GET /default.ida?NNNNNNNNNNNNNN…

Full string:

```
       GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNN

NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

NNNNNNNNNNNNNNNNNNN

%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801

%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3

%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a  HTTP/1.0
```

- **Attack description:** long URL generates a buffer overflow vulnerability in MS Internet Information Server (IIS)
- Released on 7/15/2001
- **Goal:** spread, perform DDoS against a pre-defined set of websites

Attack examples/2

Dahua webcams password reveal bug:

```
GET /current_config/passwd HTTP/1.1
Accept-Encoding: identity
X-Request: JSON
Host: 127.0.0.1:3000
X-Requested-With: XMLHttpRequest
Connection: close
User-Agent: Dahua/2.0; Dahua/3.0
```

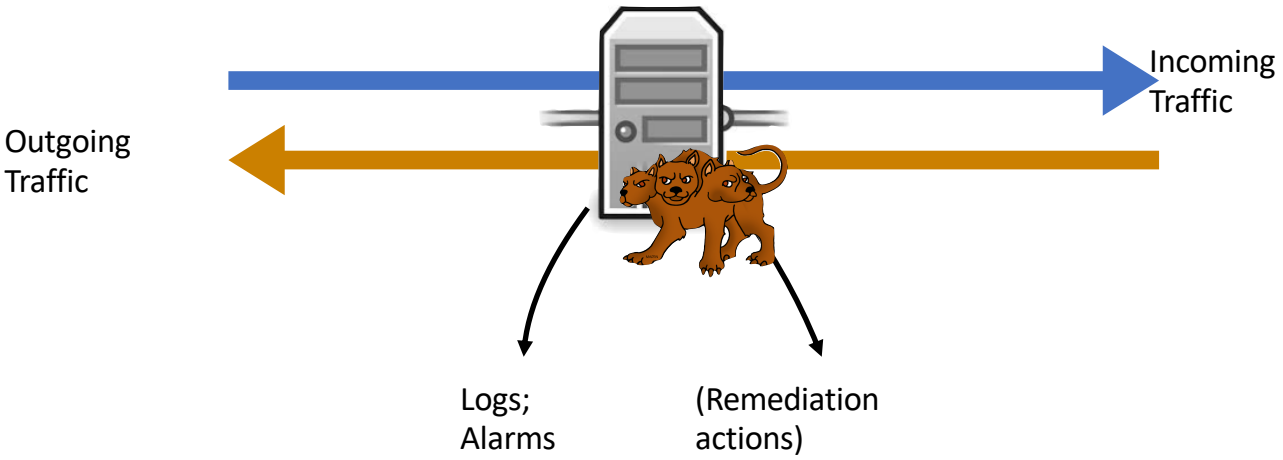
- **Attack description:** camera web interface allows download of password file
- Disclosed: 5/4/2017
- **Goal:** obtain password file to allow takeover of camera device

Approaches to Intrusion Detection/Prevention

- **Host-based:** detects the presence of malicious software agents on networked machines using various program analysis techniques (system calls, data/control flow, binary features)
- **Network-based:** detects the presence of communications with malicious purposes (e.g., exploits) and/or generated by malicious software (e.g., malware)
- **This being a networking class, we are going to focus on the latter :-)**

Intrusion Detection (Prevention) System

- "An hardware and/or software implement whose goal is to detect (and possibly prevent) network attacks by analyzing network traffic"



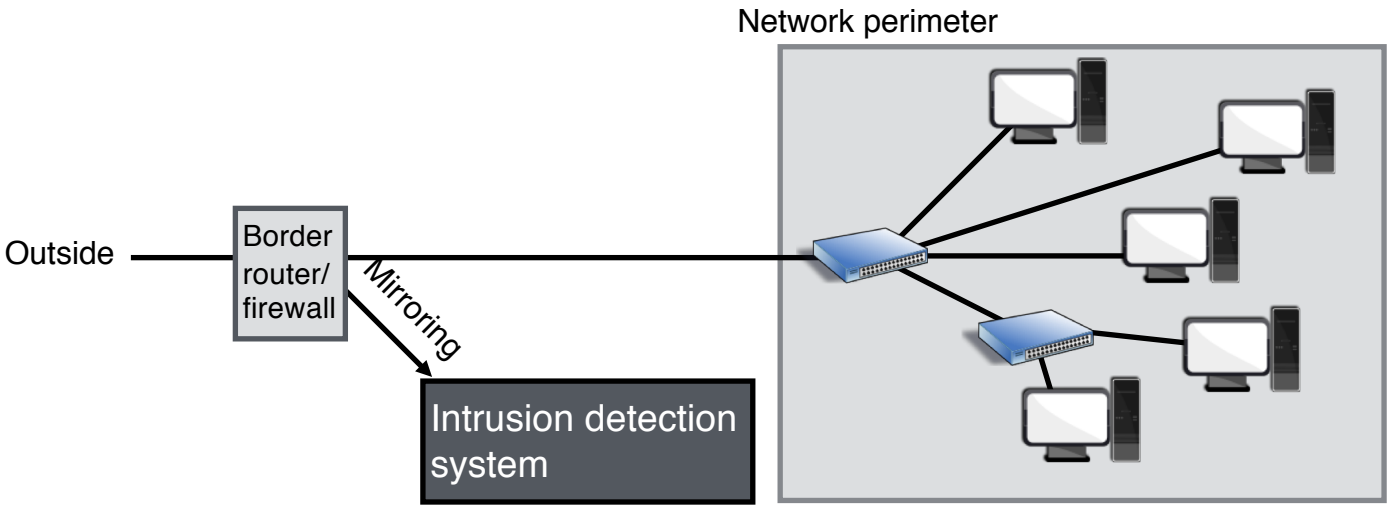
IDS (or **IPS**) for friends; will refer to it as IDS in the rest of the lecture for convenience

Deep Packet Inspection (DPI)

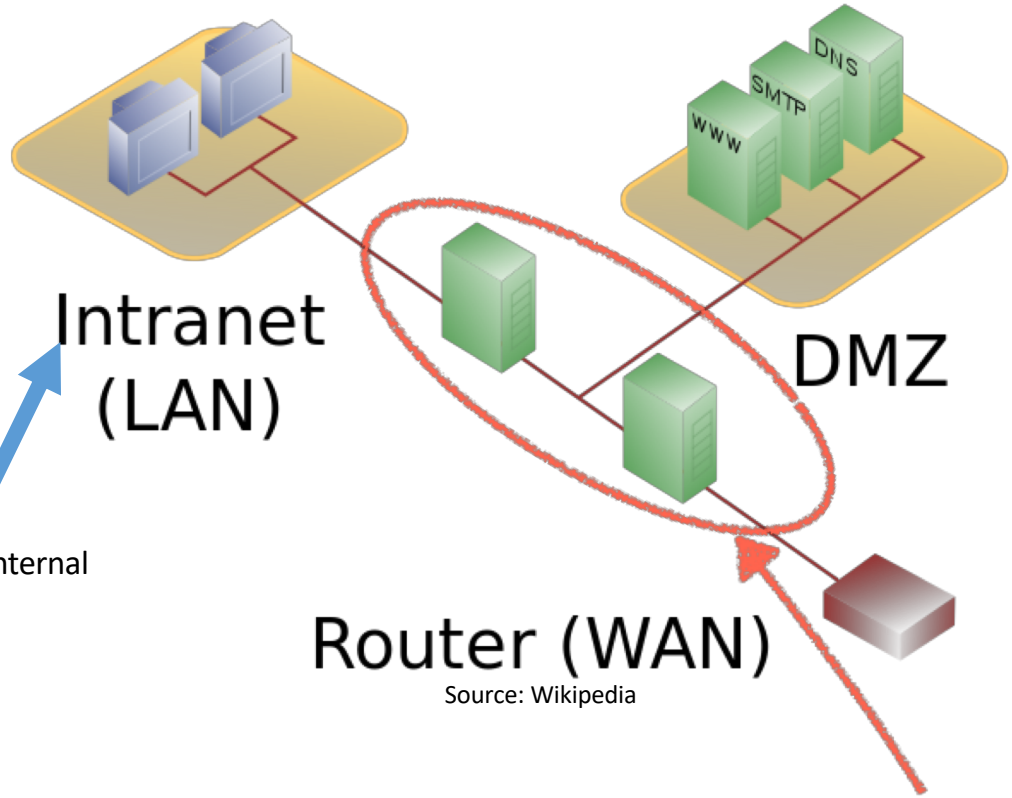
- **Deep packet inspection (DPI)** is the core operation typically performed by network-based intrusion detection systems
- It refers to the analysis of **both packet headers and payloads**
- “Deep” refers to the fact that these tools may examine the **content** of network packets, and not just header fields

IDS Placement

- Historically, IDS's have been placed on **network perimeter** (e.g. border router)



More precisely…



Source: Wikipedia

Organization's internal network

Demilitarized zone (DMZ) - hosts outward-facing servers

Possible location of an IDS

Is perimeter-only placement a good idea?

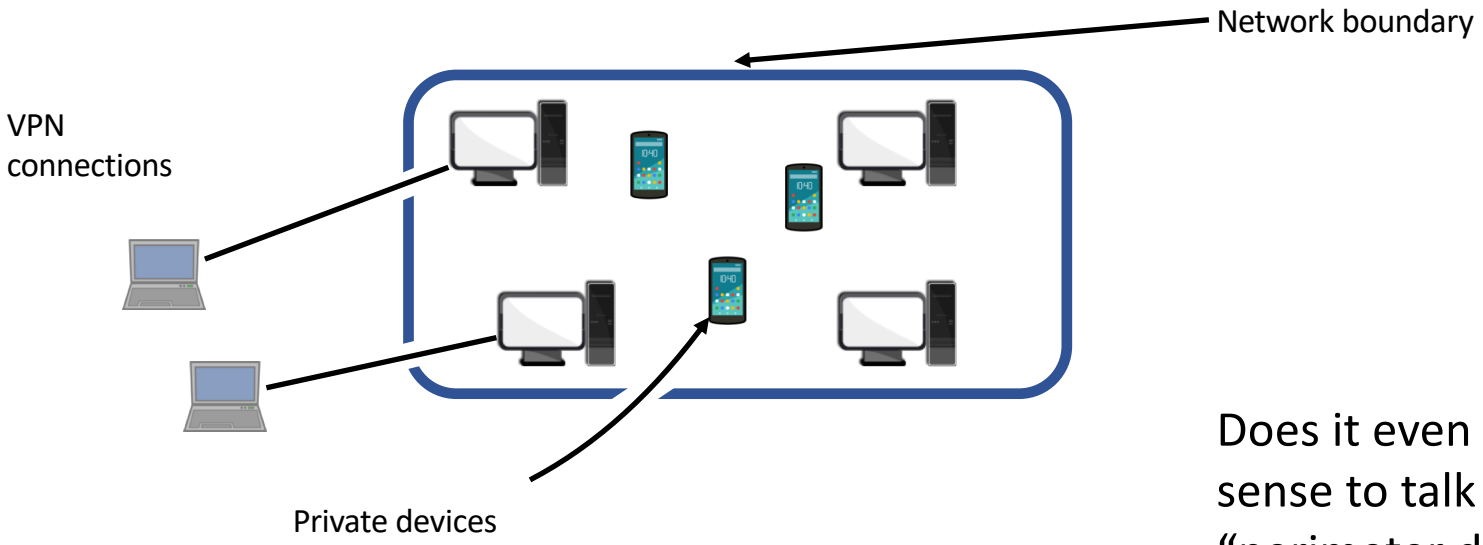
- It used to be
  - The good guys are in, the bad guys are out
- Can you guess whether this is still the case?
- Not so much anymore
  - **Problem #1:** insider threat (e.g., disgruntled employee)
  - **Problem #2:** single point of failure
    - **Once your perimeter IDS is bypassed, there is nothing you can do!**

What about placing IDS's inside the network?

- Probably a good idea
- Nowadays, necessary

Can we push this idea further?

- Observation: things like VPN, BYOD, etc. contribute to make separation between inside and outside unclear



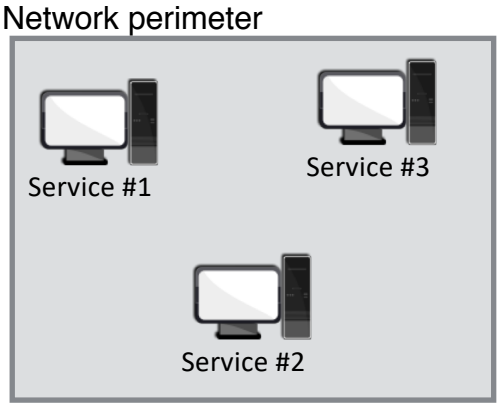
Does it even make sense to talk about "perimeter defense"?

Can we push this idea further? / 2

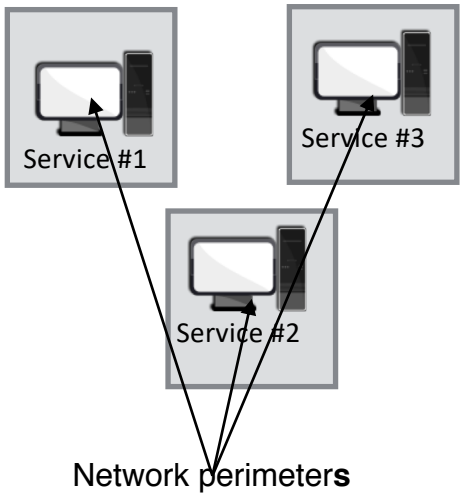
- **BeyondCorp:** the Google approach
  - “A new approach to enterprise security”, Ward & Beyer, *;login:*, 12/2014
  - Make every service accessible from the Internet, use robust authentication/encryption
  - Restrict access based on location/type of device/user
  - Does away with the notion of “internal network”

Can we push this idea further? /3

- All of the above is referred to as "**Zero Trust model**"
  - "Security perimeter" around **individual services** instead of the whole network
  - Switches from:



to:

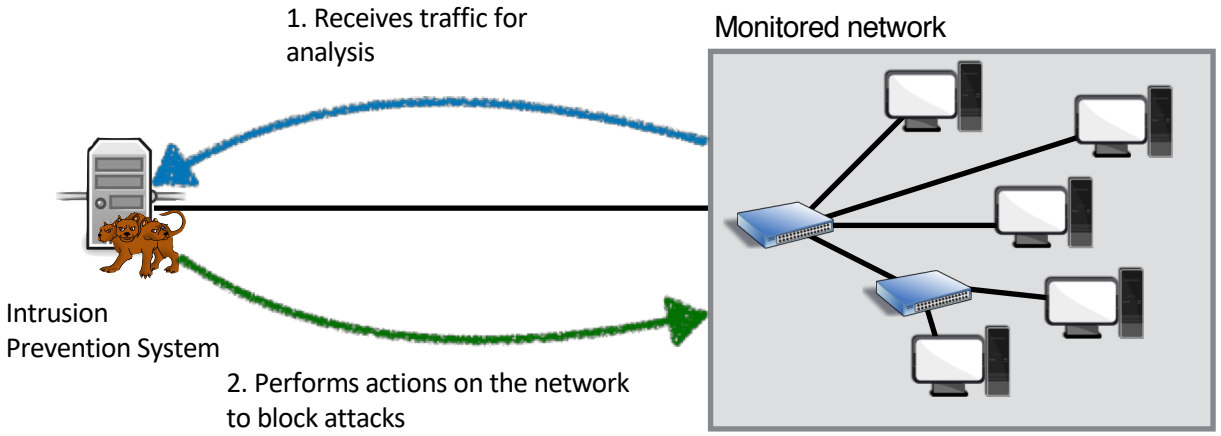


- **Too early to say if it solves all problems**

Passive vs active monitoring

- Intrusion **detection** systems focus on **passive detection**
  - **Output:** warnings to network administrators, log entries
  - Require human intervention to disrupt attacks
- Intrusion **prevention** systems focus on **active prevention**
  - **Output:** network actions that disrupt malicious actions
  - Can “take care” of (simple) attacks without human intervention

Active monitoring



- Example of **active actions?**
  - **Block TCP connections** if malware/exploits are detected
  - **Ban IP addresses** after multiple login failures
  - …
- Anything more complicated probably requires a human (Schneier, “The Future of Incident Response”)

A note about retaliation

“Hacking back is a terrible idea that just will not die” (Bruce Schneier)

- Let’s talk about why is that

Deep Packet Inspection strategies

Signature-based IDS's

- **Signature-based IDS:** perform DPI by looking for byte-level patterns in packets' payloads
- Initially, based on **fixed string matching**
  - **Example:** Code Red attack consists of constant string: *GET /default.ida?NNNNNNNNNNNNNNNN…*
- Later, extended to use **regular expressions**
  - Sommer & Paxson, "Enhancing Byte-level network intrusion detection signatures with context", CCS 2003

Signature-based IDS's/2

- **Regular expression-based attack detection**:
  - Regular expressions are used to describe the expected format of attack traffic (*signatures*)
  - Regular expressions are then converted to deterministic or non-deterministic finite automata (**DFAs** and **NFAs**), which are used to parse packets' **payloads**

Signature-based IDS's/3

- (Very old) example: certain versions of the ftpd FTP server software allow root access by changing folder to root:
  - Attack signature: `^CWD\s+~root`
  - "^" specifies string must appear at beginning of line
  - \s+ specifies that "CWD" and "root" can be separated by an arbitrary number of spaces

Signature-based IDS's/4

- Regular expressions are **more powerful than constant strings**, yet can be **matched efficiently**
- Several popular IDSs use this approach

- In reality rules are slightly more complicated - attempt to filter flows based on port, constant strings etc. before running regular expressions
- Example: full Snort rule for the FTP attack:

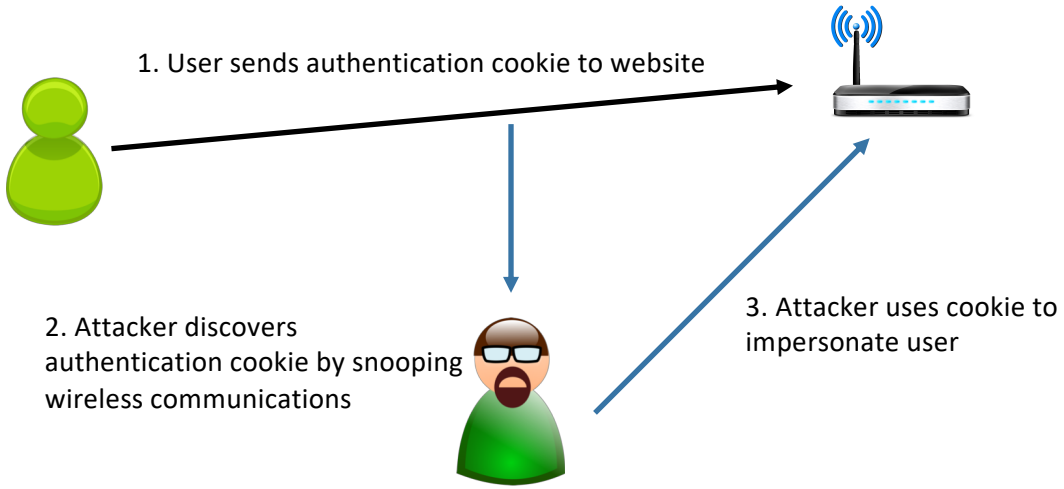
```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 ( msg:"PROTOCOL-FTP
CWD ~root attempt"; flow:to_server,established;
content:"CWD",nocase; content:"~root",distance 1,nocase;
pcre:"/^CWD\s+~root/smi";)
```

Signature-based IDS's: limitations

- **Disadvantages:**
  - **Limited:** process each connection separately (what if an attack spans multiple connections?)
  - **Fragile:** if the attacker can tweak a few bytes in the attack traffic, signatures no longer match
  - **Potentially ineffective:** many modern attacks are cannot be represented well using regular expressions

Signature-based IDS's: limitations/2

- Example attack: **HTTP session hijacking** (2010)



- **No clear signature** – why?
- Attack can only be discovered by observing that the same cookie is transmitted by two different IP addresses in a short timespan

Are we hopeless then?

- **Problem:** signature-based IDS's are easy to circumvent, and cannot detect all attacks
- (Partial) solution: **policy-based** intrusion detection systems

Bro: a policy-based IDS

- Quite revolutionary when it came out
- Several crucial **insights** that are still valid today:
  - Separate **mechanism** (how to parse traffic) from **policy** (what constitute a network attack)
    - No one-size-fit-all detection strategy (can do signature matching but other things too)
  - It is important to maintain **state** across network events
- Perform DPI by **identifying and correlating network events**

Mechanism vs Policy

- Fully understanding network traffic requires

  1. **Extracting packet payload**, and

  2. **Parsing application-level protocol** (HTTP, SMTP, etc.)

- Can't write parser for every possible network attack - they evolve too fast

- Solution:

  1. Parsing layer which abstracts traffic into flow of events

  2. Policy scripts which determines which events represent attacks

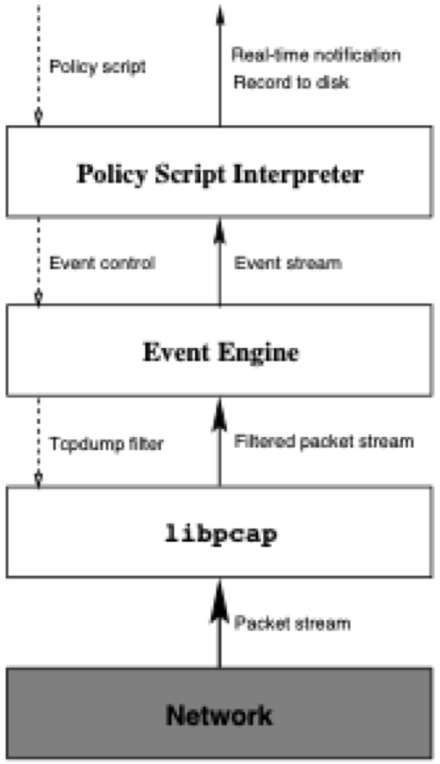


Figure 1: Structure of the Bro system

Why is this important?

- **In signature-based IDS's:**
  - The mechanism is regular expression matching over packet payloads
  - The policy is to report any expression that matches
- **In Bro:**
  - The mechanism is payload parsing/event generation
  - The policy depends on the scripts that administrator write to interpret events

Why is this important?/2

- **Example:** CVE-2002-0063: vulnerability in IPP (HTTP-based remote printing protocol)
- Protocol messages are records structured as:

Attribute name length	Attribute name	Attribute value length	Attribute value

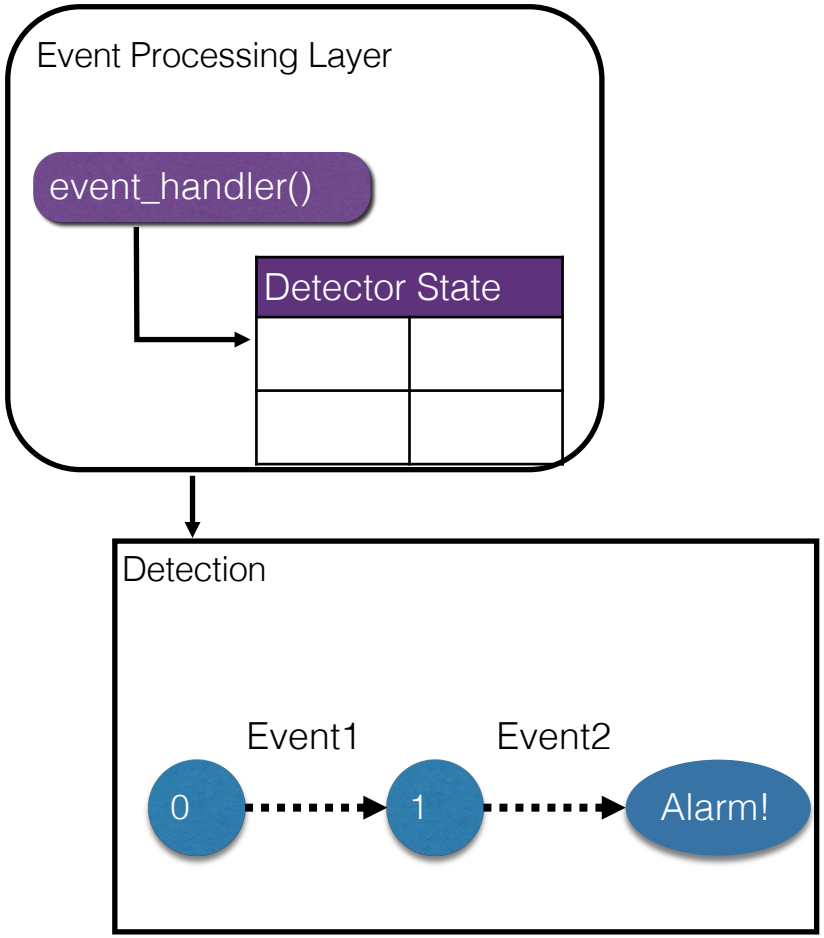
- **Attack:** send message w/ attribute name longer than 8192 bytes, trigger buffer overflow
- **Difficult to detect w/ regular expression; trivial in Bro** (by using IPP parser): if attribute name longer than 8192, report alarm

The importance of detection state

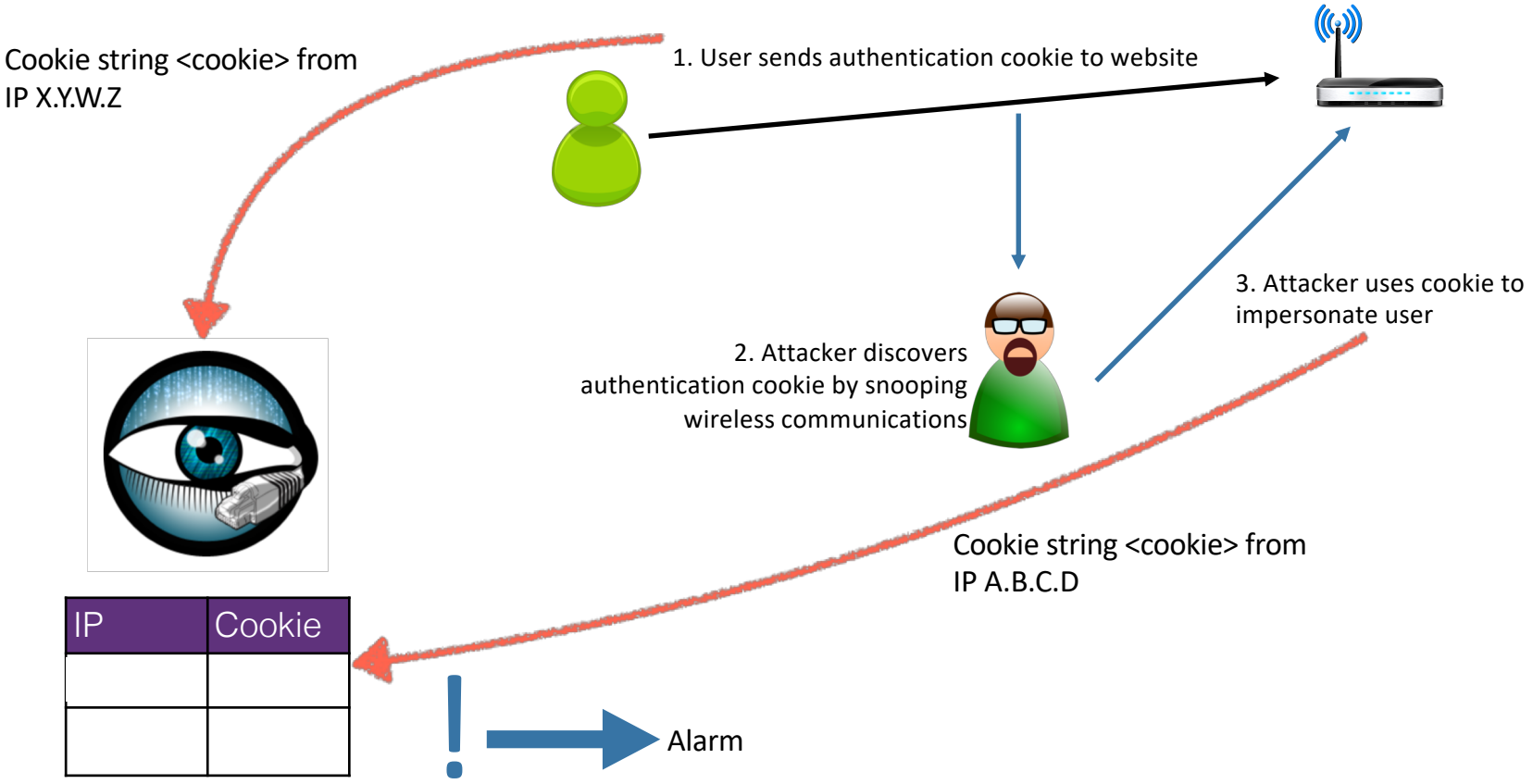
- Signature-based approach works with attacks carried over an individual connection
- What if an attack spans multiple connections?
  - Can you think of an example?
- Modern network attacks tend to span **multiple connections over time**
- Detecting them requires to maintain detection state

The importance of detection state/2

- **State in Bro:**
  - Each time a script is executed in response to a network event, it can save data in one or more tables
  - Successive executions of the script can access/update these data
  - Enables **event correlation**



The importance of detection state/3



Limitations of Bro

- **Performance**
  - Complex processing: needs to
    - Capture packets, extract payload
    - Detect application-level protocol
    - Parse application-level protocol and generate events
    - Run user scripts on events
  - Hard to parallelize or accelerate w/ specialized hardware

Limitations of Bro/2

- Compare w/ signature-based IDS's which can:
  - Process each connection independently and potentially in parallel
  - Use hardware acceleration:
    - **GPUs:** Vasiliadis et al., "Gnort: High Performance Network Intrusion Detection Using Graphics Processors", RAID 2008
    - **FPGAs:** Mitra et al., "Compiling PCRE to FPGA for accelerating Snort IDS", ANCS 2007
    - **Coprocessors:** van Lunteren and Guanella, "Hardware-Accelerated Regular Expression Matching at Multiple Tens of Gb/s" (INFOCOM 2012)