Intrusion Detection and Prevention

Tyler Bletsch
Duke University
Outline

Understanding intruders

Intrusion detection systems (IDS)

Firewalls

Intrusion prevention systems (IPS)
Two ways to categorize intruders

- **Class of intruder:** What are they after?

- **Intruder skill level:** How smart are they?
Classes of Intruders – Cyber Criminals

- Individuals or members of an organized crime group with a goal of financial reward
- Their activities may include:
  - Identity theft
  - Theft of financial credentials
  - Corporate espionage
  - Data theft
  - Data ransoming
- Typically they are young, often Eastern European, Russian, or southeast Asian hackers, who do business on the Web
- They meet in underground forums to trade tips and data and coordinate attacks
Classes of Intruders – Activists

- Are either individuals, usually working as insiders, or members of a larger group of outsider attackers, who are motivated by social or political causes.
- Also known as hacktivists.
  - Skill level is often quite low.
- Aim of their attacks is often to promote and publicize their cause typically through:
  - Website defacement
  - Denial of service attacks
  - Theft and distribution of data that results in negative publicity or compromise of their targets.
Classes of Intruders – State-Sponsored Organizations

- Groups of hackers sponsored by governments to conduct espionage or sabotage activities
- Also known as **Advanced Persistent Threats (APTs)** due to the covert nature and persistence over extended periods involved with any attacks in this class
- Widespread nature and scope of these activities by a wide range of countries from China to the USA, UK, and their intelligence allies
Classes of Intruders – Others

• Hackers with motivations other than those previously listed
• Include classic hackers or crackers who are motivated by technical challenge or by peer-group esteem and reputation
• Many of those responsible for discovering new categories of buffer overflow vulnerabilities could be regarded as members of this class
• Given the wide availability of attack toolkits, there is a pool of “hobby hackers” using them to explore system and network security
Intruder Skill Levels – Apprentice

• Hackers with minimal technical skill who primarily use existing attack toolkits

• They likely comprise the largest number of attackers, including many criminal and activist attackers

• Given their use of existing known tools, these attackers are the easiest to defend against

• Also known as “script-kiddies” due to their use of existing scripts (tools)
Intruder Skill Levels – Journeyman

- Hackers with sufficient technical skills to modify and extend attack toolkits to use newly discovered, or purchased, vulnerabilities
- They may be able to locate new vulnerabilities to exploit that are similar to some already known
- Hackers with such skills are likely found in all intruder classes
- Adapt tools for use by others
Intruder Skill Levels – Master

- Hackers with high-level technical skills capable of discovering brand new categories of vulnerabilities
- Write new powerful attack toolkits
- Some of the better known classical hackers are of this level
- Some are employed by state-sponsored organizations
- Defending against these attacks is of the highest difficulty
Intruders will want you to misapprehend their skill and class!

- Criminals may want to seem like political activists to cover their true activities.
- Apprentices want to appear like Masters.
- Masters want to appear like Apprentices.
- Etc.

- During forensics, be hesitant to jump to conclusions...
Intruder Behavior

1. Target acquisition and information gathering
2. Initial access
3. Privilege escalation
4. Information gathering or system exploit
5. Maintaining access
6. Covering tracks
(a) Target Acquisition and Information Gathering

- Explore corporate website for information on corporate structure, personnel, key systems, as well as details of specific web server and OS used.
- Gather information on target network using DNS lookup tools such as dig, host, and others; and query WHOIS database.
- Map network for accessible services using tools such as NMAP.
- Send query email to customer service contact, review response for information on mail client, server, and OS used, and also details of person responding.
- Identify potentially vulnerable services, eg vulnerable web CMS.

(b) Initial Access

- Brute force (guess) a user’s web content management system (CMS) password.
- Exploit vulnerability in web CMS plugin to gain system access.
- Send spear-phishing email with link to web browser exploit to key people.

(c) Privilege Escalation

- Scan system for applications with local exploit.
- Exploit any vulnerable application to gain elevated privileges.
- Install sniffers to capture administrator passwords.
- Use captured administrator password to access privileged information.

(d) Information Gathering or System Exploit

- Scan files for desired information.
- Transfer large numbers of documents to external repository.
- Use guessed or captured passwords to access other servers on network.

(e) Maintaining Access

- Install remote administration tool or rootkit with backdoor for later access.
- Use administrator password to later access network.
- Modify or disable anti-virus or IDS programs running on system.

(f) Covering Tracks

- Use rootkit to hide files installed on system.
- Edit logfiles to remove entries generated during the intrusion.

(Table can be found on pages 271-272 in textbook.)
Outline

Understanding intruders

Intrusion detection systems (IDS)

Firewalls

Intrusion prevention systems (IPS)
Intrusion Detection System (IDS)

- Host-based IDS (HIDS)
  - Monitors the characteristics of a single host for suspicious activity

- Network-based IDS (NIDS)
  - Monitors network traffic and analyzes network, transport, and application protocols to identify suspicious activity

- Distributed or hybrid IDS
  - Combines information from a number of sensors, often both host and network based, in a central analyzer that is able to better identify and respond to intrusion activity

Comprises three logical components:

- Sensors - collect data
- Analyzers - determine if intrusion has occurred
- User interface - view output or control system behavior
Analysis Approaches

**Anomaly detection**
- Collect data relating to the behavior of legitimate users
- Current observed behavior is compared to baseline
- Detect:
  - Denial-of-service (DoS) attacks
  - Scanning
  - Worms

**Signature/Heuristic detection**
- Scan for known malicious data patterns via signature (e.g. antivirus) or rules (e.g. ‘snort’)
- Can only identify known attacks
- Detect:
  - Reconnaissance and attacks
  - Unexpected application services
  - Policy violations
Anomaly Detection

A variety of classification approaches are used:

- **Statistical**
  - Analysis of the observed behavior using univariate, multivariate, or time-series models of observed metrics

- **Knowledge based**
  - Approaches use an expert system that classifies observed behavior according to a set of rules that model legitimate behavior

- **Machine-learning**
  - Approaches automatically determine a suitable classification model from the training data using data mining techniques
Host-Based Intrusion Detection (HIDS)

• Primary purpose is to detect intrusions, log suspicious events, and send alerts
  o Can detect both external and internal intrusions

• Data sources:
  o System call traces
  o Audit (log file) records
  o File integrity checksums
  o Registry access
### Table 8.2 Linux System Calls and Windows DLLs Monitored

#### (a) Ubuntu Linux System Calls

- accept, access, acct, adjtime, aiocancel, aioread, aiowait, aiowrite, alarm, async_daemon, auditsys, bind, chdir, chmod, chown, chroot, close, connect, creat, dup, dup2, execv, execve, exit, exportfs, fchdir, fchmod, fchown, fchroot, fcntl, flock, fork, fpathconf, fstat, fstatfs, fsync, ftime, ftruncate, getdents, getdirent, getdomainname, getdopt, getquota, getpagesize, getpeername, getpgrp, getpid, getpriority, getrlimit, getsockopt, getsockname, gotimeofday, getuid, getgid, getgroups, gethostid, gethostname, getitimer, getmsg, getpagesize, gotimeofday, getuid, gotimeofday

#### (b) Key Windows DLLs and Executables

- comctl32
- kernel32
- msvc110
- msvcr110
- mswsock
- ntdll
- ntoskrnl
- user32
- ws2_32

(Video can be found on page 280 in the textbook)
Distributed HIDS deployment

- Can put HIDS agents on many systems, manage centrally

Figure 8.2  Architecture for Distributed Intrusion Detection
Network-Based IDS (NIDS)

- **Monitors traffic** at selected points on a network
- **Examines traffic** packet by packet in real time
  - May examine network, transport, and/or application-level protocol activity
- **Comprised of:**
  - A number of sensors
  - One or more management servers
- Analysis of traffic patterns may be done at the sensor, the management server or a combination of the two
Figure 8.4 Passive NIDS Sensor
Figure 8.5 Example of NIDS Sensor Deployment
Stateful Protocol Analysis

• Understands and tracks network, transport, and application protocol **states** to ensure they progress as expected

• Higher resource use than stateless systems
Logging of Alerts

• Typical information logged by a NIDS sensor includes:
  o Timestamp
  o Connection or session ID
  o Event or alert type
  o Rating
  o Network, transport, and application layer protocols
  o Source and destination IP addresses
  o Source and destination TCP or UDP ports, or ICMP types and codes
  o Number of bytes transmitted over the connection
  o Decoded payload data, such as application requests and responses
  o State-related information
Snort: a commonly deployed NIDS

![Snort Architecture Diagram](image)

Figure 8.9 Snort Architecture

<table>
<thead>
<tr>
<th>Action</th>
<th>Protocol</th>
<th>Source IP address</th>
<th>Source Port</th>
<th>Direction</th>
<th>Dest IP address</th>
<th>Dest Port</th>
</tr>
</thead>
</table>

(a) Rule Header

<table>
<thead>
<tr>
<th>Option Keyword</th>
<th>Option Arguments</th>
<th>• • •</th>
</tr>
</thead>
</table>

(b) Options

Figure 8.10 Snort Rule Formats
**Table 8.3**
*Snort Rule Actions*

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>Generate an alert using the selected alert method, and then log the packet.</td>
</tr>
<tr>
<td>log</td>
<td>Log the packet.</td>
</tr>
<tr>
<td>pass</td>
<td>Ignore the packet.</td>
</tr>
<tr>
<td>activate</td>
<td>Alert and then turn on another dynamic rule.</td>
</tr>
<tr>
<td>dynamic</td>
<td>Remain idle until activated by an activate rule, then act as a log rule.</td>
</tr>
<tr>
<td>drop</td>
<td>Make iptables drop the packet and log the packet.</td>
</tr>
<tr>
<td>reject</td>
<td>Make iptables drop the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.</td>
</tr>
<tr>
<td>sdrop</td>
<td>Make iptables drop the packet but does not log it.</td>
</tr>
</tbody>
</table>
ALSO: Virtual Machine Introspection

- Look at a VM from the outside

Challenge: The Semantic Gap
Virtual Machine Introspection

- Examples: libVMI, VMware vShield Endpoint, etc.
Honeypots

- Decoy systems designed to:
  - Lure a potential attacker away from critical systems
  - Collect information about the attacker’s activity
  - Encourage the attacker to stay on the system long enough for administrators to respond

- Systems are filled with fabricated information that a legitimate user of the system wouldn’t access

- Resources that have no production value
  - Therefore incoming communication is most likely a probe, scan, or attack
  - Initiated outbound communication suggests that the system has probably been compromised

- Classified as being either low or high interaction
  - Low interaction honeypot consists of a software package that emulates particular IT services or systems well enough to provide a realistic initial interaction, but does not execute a full version of those services or systems
  - High interaction honeypot is a real system, with a full operating system, services and applications, which are instrumented and deployed where they can be accessed by attackers
Outline

Understanding intruders

Intrusion detection system (IDS)

Firewalls

Intrusion prevention systems (IPS)
Design goals

All traffic from inside to outside, and vice versa, must pass through the firewall.
Only authorized traffic as defined by the local security policy will be allowed to pass.
The firewall itself is immune to penetration.
# Types of firewalls

## Simpler, less expressive, less resource-intensive

<table>
<thead>
<tr>
<th>Type</th>
<th>Logic</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet filter</td>
<td>Decide on per-packet basis</td>
<td>• Simple</td>
<td>• Dumb</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fast</td>
<td>• Not very expressive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Easy to configure</td>
<td></td>
</tr>
<tr>
<td>Stateful packet inspection</td>
<td>Decide on stream or higher level basis</td>
<td>• More expressive</td>
<td>• More resource intensive</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• More configuration</td>
</tr>
<tr>
<td>Application-level proxy</td>
<td>Understands app-level traffic</td>
<td>• Can enforce app-relevant restrictions</td>
<td>• Need one customized for each app</td>
</tr>
</tbody>
</table>

## More complex, more expressive, more resource-intensive
Placement of firewalls (1)

LAN firewall
Host-based firewall
Firewall Filter Characteristics

Characteristics that a firewall access policy could use to filter traffic include:

- **IP address and protocol values**
  - This type of filtering is used by packet filter and stateful inspection firewalls
  - Typically used to limit access to specific services

- **Application protocol**
  - This type of filtering is used by an application-level gateway that relays and monitors the exchange of information for specific application protocols

- **User identity**
  - Typically for inside users who identify themselves using some form of secure authentication technology

- **Network activity**
  - Controls access based on considerations such as the time or request, rate of requests, or other activity patterns
• Book spends a long time on this, but it’s simple: firewalls have human-built rules \textit{and} can only deal with packets that go through them.

• Two scenarios they don’t help:
  - HTTP service has a vulnerability and firewall allows HTTP
  - Firewall is at ISP uplink but rogue cell phone gets inside of LAN via WiFi
Packet Filtering Firewall

- Applies rules to each incoming and outgoing IP packet
  - Typically a list of rules based on matches in the IP or TCP header
  - Forwards or discards the packet based on rules match

Filtering rules are based on information contained in a network packet

- Source IP address
- Destination IP address
- Source and destination transport-level address
- IP protocol field
- Interface

- Two default policies:
  - Discard - prohibit unless expressly permitted
    - More conservative, controlled, visible to users
  - Forward - permit unless expressly prohibited
    - Easier to manage and use but less secure
# Table 9.1

Packet-Filtering Examples

<table>
<thead>
<tr>
<th>Rule</th>
<th>Direction</th>
<th>Src address</th>
<th>Dest address</th>
<th>Protocol</th>
<th>Dest port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In</td>
<td>External</td>
<td>Internal</td>
<td>TCP</td>
<td>25</td>
<td>Permit</td>
</tr>
<tr>
<td>2</td>
<td>Out</td>
<td>Internal</td>
<td>External</td>
<td>TCP</td>
<td>&gt;1023</td>
<td>Permit</td>
</tr>
<tr>
<td>3</td>
<td>Out</td>
<td>Internal</td>
<td>External</td>
<td>TCP</td>
<td>25</td>
<td>Permit</td>
</tr>
<tr>
<td>4</td>
<td>In</td>
<td>External</td>
<td>Internal</td>
<td>TCP</td>
<td>&gt;1023</td>
<td>Permit</td>
</tr>
<tr>
<td>5</td>
<td>Either</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Any</td>
<td>Deny</td>
</tr>
</tbody>
</table>
Stateful Inspection Firewall

Tightens rules for TCP traffic by creating a directory of outbound TCP connections

- There is an entry for each currently established connection
- Packet filter allows incoming traffic to high numbered ports only for those packets that fit the profile of one of the entries in this directory

Reviews packet information but also records information about TCP connections

- Keeps track of TCP sequence numbers to prevent attacks that depend on the sequence number
- Inspects data for protocols like FTP, IM and SIPS commands
## Table 9.2

Example Stateful Firewall Connection State Table

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Source Port</th>
<th>Destination Address</th>
<th>Destination Port</th>
<th>Connection State</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.100</td>
<td>1030</td>
<td>210.9.88.29</td>
<td>80</td>
<td>Established</td>
</tr>
<tr>
<td>192.168.1.102</td>
<td>1031</td>
<td>216.32.42.123</td>
<td>80</td>
<td>Established</td>
</tr>
<tr>
<td>192.168.1.101</td>
<td>1033</td>
<td>173.66.32.122</td>
<td>25</td>
<td>Established</td>
</tr>
<tr>
<td>192.168.1.106</td>
<td>1035</td>
<td>177.231.32.12</td>
<td>79</td>
<td>Established</td>
</tr>
<tr>
<td>223.43.21.231</td>
<td>1990</td>
<td>192.168.1.6</td>
<td>80</td>
<td>Established</td>
</tr>
<tr>
<td>219.22.123.32</td>
<td>2112</td>
<td>192.168.1.6</td>
<td>80</td>
<td>Established</td>
</tr>
<tr>
<td>210.99.212.18</td>
<td>3321</td>
<td>192.168.1.6</td>
<td>80</td>
<td>Established</td>
</tr>
<tr>
<td>24.102.32.23</td>
<td>1025</td>
<td>192.168.1.6</td>
<td>80</td>
<td>Established</td>
</tr>
<tr>
<td>223.21.22.12</td>
<td>1046</td>
<td>192.168.1.6</td>
<td>80</td>
<td>Established</td>
</tr>
</tbody>
</table>
Figure 9.2  Example Firewall Configuration

- Web server(s)
- Email server
- DNS server

- Application and database servers
- Workstations

- Internal DMZ network

- Internal protected network

- External firewall
- LAN switch

- Internet

- Boundary router
- LAN switch
- Internal firewall
Outline

Understanding intruders

Intrusion detection systems (IDS)

Firewalls

Intrusion prevention systems (IPS)
Intrusion Prevention System (IPS):
It’s IDS that can do something about stuff it sees
Host-Based IPS (HIPS)

• Can make use of either signature/heuristic or anomaly detection techniques to identify attacks
  • Signature: focus is on the specific content of application network traffic, or of sequences of system calls, looking for patterns that have been identified as malicious
  • Anomaly: IPS is looking for behavior patterns that indicate malware

• Examples of the types of malicious behavior addressed by a HIPS include:
  • Modification of system resources
  • Privilege-escalation exploits
  • Buffer-overflow exploits
  • Access to e-mail contact list
  • Directory traversal
The Role of HIPS

- Many industry observers see the enterprise endpoint, including desktop and laptop systems, as now the main target for hackers and criminals
  - Thus security vendors are focusing more on developing endpoint security products
  - Traditionally, endpoint security has been provided by a collection of distinct products, such as antivirus, antisyware, antispam, and personal firewalls
- Approach is an effort to provide an integrated, single-product suite of functions
  - Advantages of the integrated HIPS approach are that the various tools work closely together, threat prevention is more comprehensive, and management is easier
- A prudent approach is to use HIPS as one element in a defense-in-depth strategy that involves network-level devices, such as either firewalls or network-based IPSs
Network-Based IPS (NIPS)

- Inline NIDS with the authority to modify or discard packets and tear down TCP connections
- Makes use of signature/heuristic detection and anomaly detection
- May provide flow data protection
  - Requires that the application payload in a sequence of packets be reassembled
- Methods used to identify malicious packets:
  - Pattern matching
  - Stateful matching
  - Protocol anomaly
  - Traffic anomaly
  - Statistical anomaly
**Snort Inline**

- Enables Snort to function as an intrusion prevention system
- Includes a replace option which allows the Snort user to modify packets rather than drop them
  - Useful for a honeypot implementation
  - Attackers see the failure but cannot figure out why it occurred

<table>
<thead>
<tr>
<th>Drop</th>
<th>Reject</th>
<th>Sdrop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snort rejects a packet based on the options defined in the rule and logs the result</td>
<td>Packet is rejected and result is logged and an error message is returned</td>
<td>Packet is rejected but not logged</td>
</tr>
</tbody>
</table>
NIPS at Duke

• All the “Is this your student?” emails I’ve gotten from OIT were from Duke’s IDS/IPS system, which is comprised of several components

• Examples:
  
  ▪ Portscans are detected using a homespun python script that looks at flow data from a network logger and triggers if unique targets for a given service exceeds a threshold – threshold is configurable per service.
    
    • Example alert data:
      
      The alert condition for 'Duke Scanners by IP' was triggered.
      
      This alert triggers when the argus scanner detect processes detects an IP on our networks that appears the be scanning. The behavior should be investigated to make sure that it was intentional and not malicious. If so and is likely to reoccur, we should see if the IP is static and possibly exclude it from this alert.
      
      ip,port,hosts_touched,threshold,firstseen,lastseen,host

  ▪ Auto-blocking of VictimCo incoming IP address: Caused because the unencrypted reverse shell content contained info reading an .htaccess and/or .htpasswd file (one of many rules that this flow would eventually violate)
    
    • “Solved” by whitelisting VictimCo with OIT’s IDS/IPS systems
# Conclusion

## Understanding intruders
- Criminal/activist/state/other
- Skill level

## Intrusion detection systems (IDS)
- Look for anomalies or signatures, log/alert accordingly
- Either host-based or network-based

## Firewalls
- Block traffic based on rules

## Intrusion prevention system (IPS)
- It’s an IDS but it takes action