Firewalls and intrusion detection systems

TkL Markus Peuhkuri

2008-04-29

Lecture topics

- Firewalls
- Security model with firewalls
- Intrusion detection systems
- Intrusion prevention systems
- How to prevent and detect attacks
- After this lecture, you should
  - be able to enumerate different firewall types
  - know what firewalls are good for
  - know IDS, IPS and honeypots

What is a firewall

- Divides network into two (or more) parts with *different security policy*
  - internal network ⇔ Internet
  - engineering ⇔ accounting: the other network need not be a less secure one that the other one. They just have different security policies or different assets to protect.
  - internal network ⇔ public servers ⇔ Internet
  - building automation ⇔ VoIP ⇔ surveillance system
- Enforces security policy
  - allowed traffic
  - prohibited traffic

  Refer to IPSec security policy database (SPD): traffic is bypassed, discarded, or bypassed as protected.
- May have additional roles, such as a VPN endpoint

Firewall types

Packet-filtering makes decision based only packet fields

- router ACL (access control list)
- TCP implicit state: for example to disallow incoming connections, firewall will drop any packet that has SYN flag set but no ACK and allows any packet with SYN+ACK.
- difficult with UDP, also some other TCP-based protocols such as FTP in active mode, where server establishes connection to client.

Stateful keeps track on connections

- maintains connection state

  - single point of failure
  - has to have some timeout mechanism as the state space is limited. Some attacks may exhaust state space.
    ⇒ random disconnections

- possible to accept related connections: some protocols need an application gateway.

Application gateway interpret connection on application level

- checks if application traffic is valid
- protects from a simple port changes like running ssh protocol on port 443 (https).
- may provide a payload inspection to detect malicious payload
- proxy servers
  - call-out
  - in-line (transparent)

Firewall types

Address-translation between internal numbering and external addresses

- using NAPT provides same security as prohibiting incoming TCP and UDP
- internal topology can be hidden

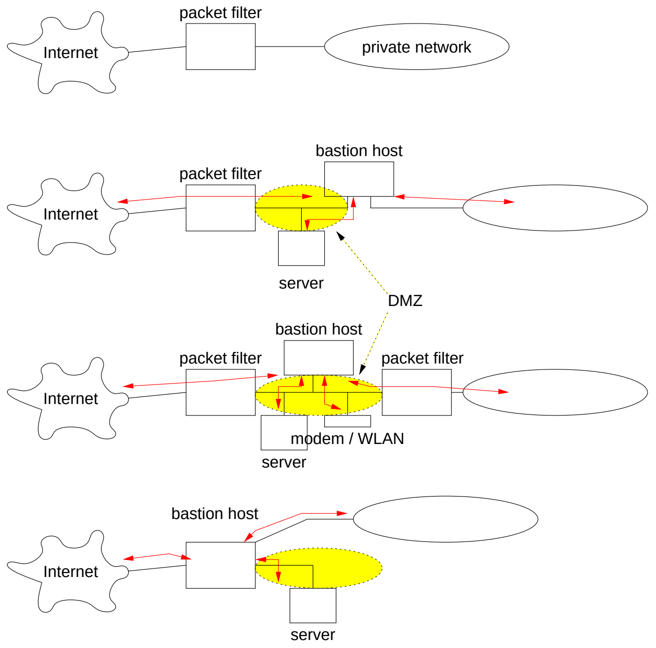
Host-based or software firewalls add on application security

- completes application security and access control
- possibly user- and application-level control

Hybrid use combination of different types for performance

- check start of connection with application gateway, switch to stateful filtering
  ⇒ better performance as the bulk of traffic is handled by the fast path.

Firewall topologies



Building firewall rules

- Defining default policy
  - "everything not prohibited is allowed"
    - "router" ACL
    - enumerate vulnerable services and protect them
  - "everything not allowed is prohibited"
    - enumerate need and safe services and allow only those
  - both policies need continuous updating
- There should be only one rule matching for each packet
  - multiple overlapping rules
  - order of rules matters
  - performance issues: hardware-based routers/firewalls can handle certain number of rules without significant performance penalty. For software-based firewalls order of rules does matter.
- Possibility to oversight
- High-level specification languages are not a solution

Deploying multiple firewalls

- Helps to limit the impact of attack
- Protection by diversity
  - on the other hand, multiple systems to update
- Designing rules even more complicated

What firewall protects and what not

- Protects
  - from known, vulnerable protocols
  - static network configuration
- Does *not* protect for / from
  - executable/active content, unless has integrated virus scanner that detects it, often targeted attacks go undetected
  - malicious insider
  - loopholes: modems, WLAN, mobile networks
  - carry-in/out attacks such as notebooks, mass storage, rogue WLAN APs
  - new attacks using applications previously considered safe
  - most DoS attacks
- May result a "hard perimeter, mellow inside"
  - failure to update internal systems
  - selecting insecure protocols and applications

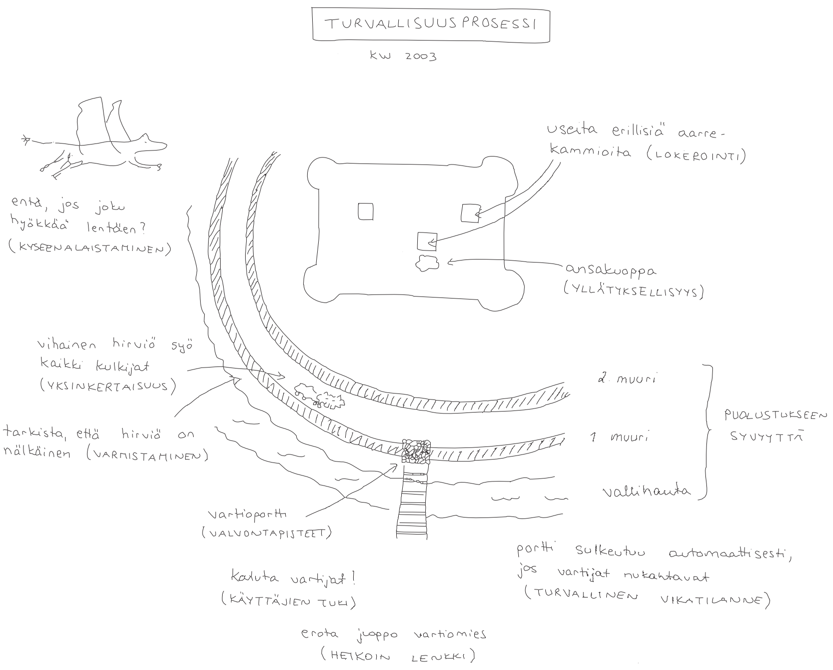
Problems caused for applications

- Firewall should be transpartent for legimate traffic

  > The introduction of a firewall and any associated tunneling or access negotiation facilities *MUST NOT* cause unintended failures of *legitimate* and *standards-compliant* usage that would work were the firewall not present.[2]

- Often fails badly for non-http applications
  - SIP: you need to open right ports based on signaling
  - ALG (Application Level Gateway): support function in firewall to undestand application
    - needs to be done for each application and each firewall
  - midcom (MIDdlebox COMmunication)
    - MIDCOM agents: ALGs external for middlebox
⇒ faster updates
    - communicates with firewall or other middlebox

Security in organisation


TURVALLISUUSPROSESSI
KW 2003
useita erillisiä aarre-kammioita (LOKEROINTI)
entä, jos joku hyökkää lentäen? (KYSEENALAISTAMINEN)
ansakuoppa (YLLÄTYKSELLISYYS)
vihainen hirviö syö kaikki kulkijat (YKSINKERTAISUUS)
tarkista, että hirviö on nälkäinen (VARMISTAMINEN)
2. muuri
1 muuri
vallihauta
PUOLUSTUKSEEN SYVYYTTÄ
vartioportti (VALVONTAPISTEET)
portti sulkeutuu automaattisesti, jos vartijat nukahtavat (TURVALLINEN VIKATILANNE)
koulута vartijat! (KÄYTTÄJIEN TUKI)
erota juoppo vartiomies (HEIKOIN LENKKI)

How secure are firewalls

- Common Vulnerabilities and Exposures: 237 matches on "firewall" `http://cve.mitre.org/cve/`

Check Point FireWall-1	31 entries
Cisco	32 entries
Juniper	1 entry
Linux	17 entries
Nokia	2 entries
Norton	12 entries
Symantec	30 entries
WatchGuard	12 entries

- More features (VPN, virus checks, QoS protection)
⇒ more code
⇒ more bugs
⇒ more vulnerabilities

Intrusion Detection Systems

- How to make sure that the firewall is not leaking
  - rule-based
  - anomaly-based
- How to detect internal attacks
- IDS is designed to
  - detect,
  - identify, and
  - report malicious activity
- IDS can be located different places
  - application
  - host
  - network

Application and host IDS

- An application instrumented to identify abnormal actions
  - high level of abstraction
  - user actions monitored
  - policy violations
  - application log analysis
  - access to encrypted data
  - may not protect from application flaws
- Host instrumented
  - reference monitor
  - actions by a user and an application
  - host log analysis
- Log analysis best done on separate host
  - provides after-the-fact analysis
  - vulnerable to network attacks DoS on log server
  - messages transmitted in clear unless IPSec is deployed

Network IDS

- Monitors traffic
  - best done with signal splitters operating on physical layer
- Large volume of data
  - low level of abstraction
  - encrypted traffic problematic
- Mostly misuse detection
  - recorded patterns of misuse (signatures)
  - frequent updates (like virus scanners)

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22
 ( msg:"EXPLOIT ssh CRC32 overflow /bin/sh";
   flow:to_server,established;
   content:"/bin/sh";  )
```

Network IDS...

- Anomaly detection
  - detecting differences to normal
    - threshold detection
    - statistical profile
    - rule-based detection
  - learning system
- Large number of alerts: an example
  - 3700 alerts from corporate network per day
  - 48 should be studied in detail
  - 2 warrant an action

IDS in large network

- Should one monitor on every link
  ⇒ very expensive
- Select important links
  - full census on those
- Do random sampling on other links
  - if one samples every 512th packet and sends it to a central location
    ⇒ not a big increase in traffic
  - large problems notified immediately

Honeypots

- A false system similar to production system
  - all access illegal
    ⇒ any accessing is a potential intruder
- Used as part of IDS
  - a connection results detailed monitoring
- How to keep an attacker from telling the difference from a real system
  - should be not too weak
  - should have “real” data and traffic
  - if a virtual system, should not be visible. Virtual systems (like VMWare, Xen and other) have benefit for external monitoring outside of honeypot OS. However, there are glues an attacker can use to identify being in honeypot. That may not be serious, however, because many production environments are now run virtualised.

IDS reaction too slow

- IDS identifies attack and send alert
  - analysis may not be real-time
  - corrective actions may take time
- Epidemic security problem may require instant actions [6]
- A system can be scanned, attacked, and compromised in a minute or less
  ⇒ Need for an automatic security system

Intrusion Prevention Systems (IPS)

- IDS with an automatic response
- Suffers from a large number of false alerts
  ⇒ may result denial of service
- A firewall with automatic ACL update
- Virus scanners are host-based IPS
- Still at early stages
  - does not stop vendors from marketing…

Traffic traceback

- Problem: where incoming attack traffic originates
- Source IP cannot be trusted
  - sender can put it to any address
  - ingress filtering not deployed universally [1]
- Should not need additional hardware or load on routers
- Scalability problems, few proposals [3, 4, 5]

Security in Ad-hoc networks

- Ad-hoc networks an interesting topic
  - self-building topology
  - extending network coverage
- Must rely on the other hosts
  - no central authority, block lists
  - no trusted core network
  - routing done by devices
- Public key-based per-packet authentication too heavy
  - modern PC throughput few ten kbit/s, much less for battery-powered device
- How to communicate trustfulness?

Challenges in All-IP world

- Large number of non-technical users
  - the "`--:--`" generation
  - rightful ignorance: I want to watch movies — fixing security problems does not match to my idea of relaxing.
- Service provider responsibility
- Multi-vendor environment

Summary

- Firewall and IDS are good tools
- Must know their limitations
- Future challenges
  - accurate detection of malicious activity
  - security in ubiquitous computing
  - trust in autonomous systems
  - providing security for couch potatoes

References

[1] J. Case, R. Mundy, D. Partain, and B. Stewart. *Introduction to Version 3 of the Internet-standard Network Management Framework*, April 1999. RFC 2570. URL:`http://www.ietf.org/rfc/rfc2570.txt`.

[2] N. Freed. *Behavior of and Requirements for Internet Firewalls*, October 2000. RFC 2979. URL:`http://www.ietf.org/rfc/rfc2979.txt`.

[3] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for IP traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, August 2000. An early version of the paper appeared as techreport UW-CSE-00-02-01 available at: `http://www.cs.washington.edu/homes/savage/traceback.html`.

[4] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. Hash-Based IP traceback. In Roch Guerin, editor, *Proceedings of the ACM SIGCOMM 2001 Conference (SIGCOMM-01)*, volume 31, 4 of *Computer Communication Review*, pages 3–14, New York, August 27–31 2001. ACM Press.

[5] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Beverly Schwartz, Stephen T. Kent, and W. Timothy Strayer. Single-packet ip traceback. *IEEE/ACM Trans. Netw.*, 10(6):721–734, 2002.

[6] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium (Security '02)*. To be appear. URL:`http://www.cs.berkeley.edu/~nweaver/cdc.web/`.