Netzwerksicherheit A WS 2004/2005

Emulieren von Angriffen

Thomas Drecker, 200028757
Stefan Dierichs, 200124365

Inhaltsverzeichnis

1 Einleitung

Durch die Vernetzung von Computern durch das Internet sind potentiellen Angreifern viele Möglichkeiten geboten. Sicherheitslücken (problemen) können ausgenutzt werden um diese Systeme zu attackieren. Sicher Netzwerke sollen Computersysteme soweit wie mögliche vor der Beeinflussung durch Angreifer schützen. Laut einer Studie von CompTia über Sicherheitsheitsprobleme in Netzwerken sind 80% der Netzwerke von Organisationen und Firmen mit Computerviren und –würmern befallen. 65% der Netzwerke hatten über das Netzwerk kompromittierte Systeme und 33% waren im Jahre 2002 Opfer von Denial of Service [DoS] .

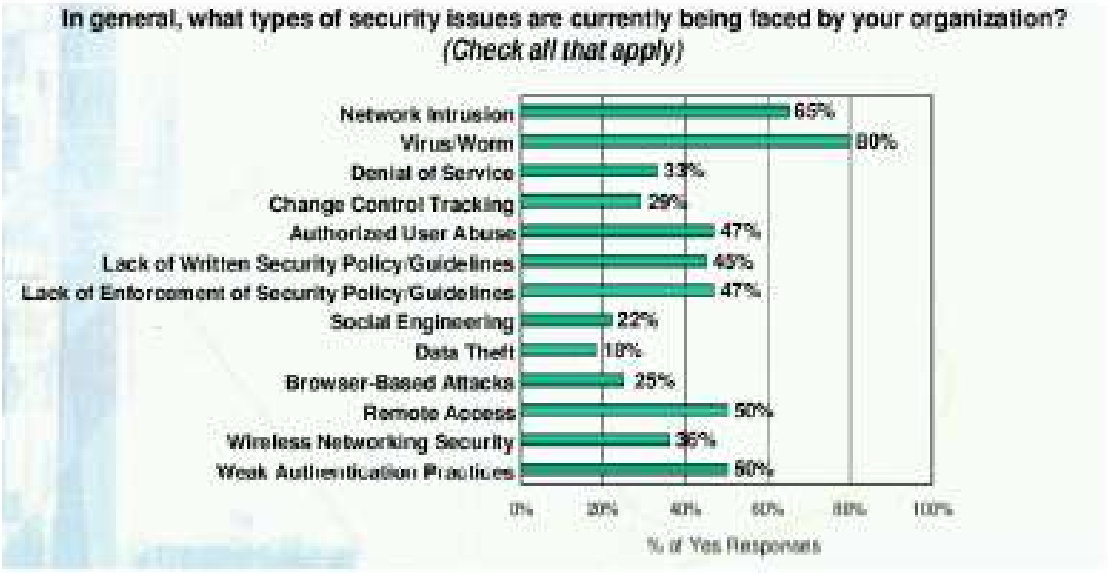


Abbildung 1: compTIA IT Security Study 2002

Die aktuelle Studie der CompTia 2004 nennt neben fehlenden Sicherheitskonzepten, menschliches Fehlverhalten als eine noch häufigere Hauptursache für Sicherheitsverletzungen im IT-Bereich. Bessere Schulung und Vorbereitung der verantwortlichen Mitarbeiter bewirkten jedoch, dass die Auswirkungen der Verletzungen auf den Geschäftsbetrieb begrenzt werden konnten.
Im Hinblick auf diese Studien ist es ersichtlich, dass man sich am besten Schützen kann, wenn man sich der Gefahr bewusst ist oder noch besser, wenn man die Gefahr kennt.

2 Emulieren von Angriffen

2.1 Emulieren und Simulieren

Wenn wir von Emulation oder Simulation sprechen, sollte man sich erst noch einmal vor Augen führen, was die einzelnen Begriffe bedeuten.
Spricht man von einer Simulation, so geht man von einem simulierenden System und einem Simulationsmodell aus. Das Simulationsmodell stellt eine Abstraktion des zu simulierenden Systems dar.
Gründe für eine Simulation sind :

- Die Untersuchung am realen System wäre zu aufwändig, zu teuer, ethisch nicht vertretbar oder zu gefährlich.
- Das reale System existiert noch gar nicht.
- Das reale System lässt sich nicht direkt beobachten
- Für Experimente kann ein Simulationsmodell wesentlich leichter modifiziert werden, als das reale System.

Die Folgen sind also nicht *wirklich*, und der Simulator ist nicht geeignet, das nachgeahmte Original in mehr als nur Teilaspekten zu ersetzen.

Eine Emulation hingegen wird in der Computertechnik als das funktionelle Nachbilden eines Systems durch ein anderes bezeichnet. Das nachbildende System enthält die gleichen Daten, führt die gleichen Programme aus und erzielt die gleichen Ergebnisse wie das nachgebildete System.
Anwendungsbereiche einer Emulation sind:

- Software für andere Systeme zu entwickeln und zu testen
- Ein Emulator erlaubt es, sich in Systeme einzuarbeiten, deren Anschaffung sonst sehr aufwändig wäre.

Beispiele hierfür sind Prozessor- und Betriebssystememulatoren, durch die Programme auf Plattformen ausgeführt werden können, für die sie ursprünglich nicht gedacht waren; der Rechner gibt also vor, ein anderer zu sein.

Wir sprechen in unserem Vortrag von der „Emulation von Netzangriffen“ da es sich bei der Nutzung der verschiednen Tools nicht um simulierten Verkehr handelt, sondern um real erzeugten Verkehr und um real durchgeführte Angriffe handelt.

2.2 Ergebnisse eines Emulierten Angriffes

Die Durchführung von Angriffsemulationen hat immer ein bestimmstes ziel vor Augen. Da die Emulation recht aufwendig sein kann, sollte man die Ziele und Ergnisse vorab definieren.

Die Ergebnisse eines emulierten Angriffes können wie folgr beschrieben werden:

- Ziel eines emulierten Angriffes kann es sein, die Auswirkung von Angriffen auf die Netzinfrastruktur zu testen. Dabei erhält der Netzwerkadministrator Informationen, wie abfällig die Netzinfrastruktur gegenüber Angriffen ist.
- Der Netzwerkverantworliche kann das sicherheitslevel der eingesetzten Sicherheitsanwendung kann bestimmt werden. Man erlangt Ergebnisse, wie gut man gegen Angriffe gerüstet ist.
- Ferner kann es von Interesse sein, Statistik über Angriffe und Angriffsversuche zu erstellen. Um somit das Verhalten und die verwendeten Angriffe zu studieren.

2.3 Aufbau zur Emulation von Angriffen

Die Emulation von Angriffen benötigt eine bestimmte Infrastruktur um Angriffe nachbilden zu können. Das folgende Bild zeigt einen möglichen Aufbau der Netzstruktur für eine Angriffsemulationen.

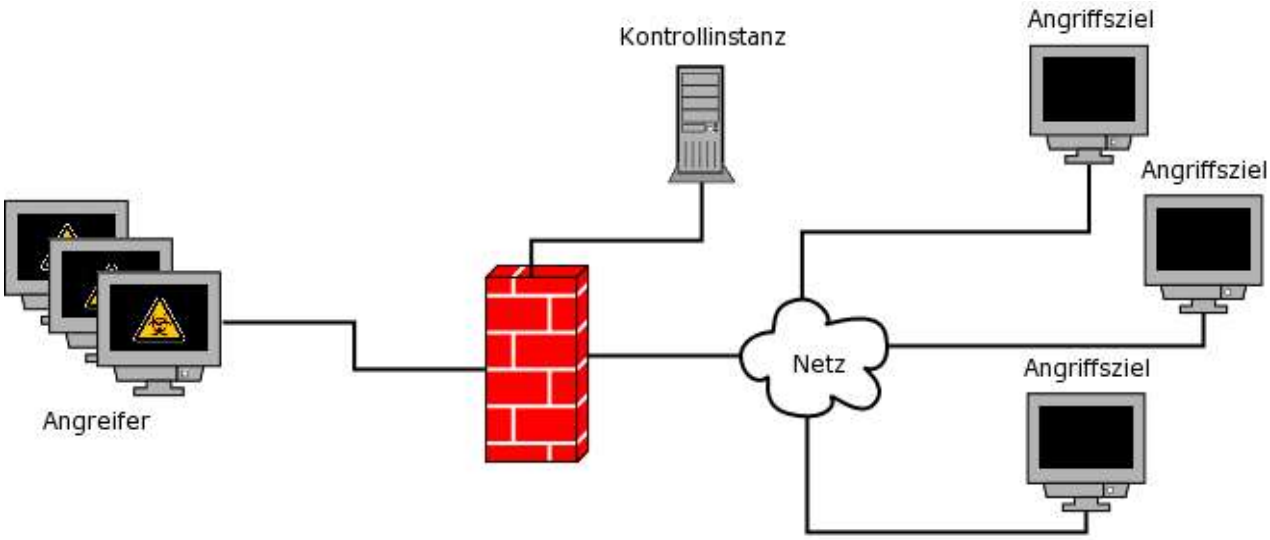


Abbildung 2: Aufbau zur Emulation von Angriffen

Der Aufbau berücksichtigt den Fall, dass ein Angriff meist aus einem unsicheren Netz (z.B. Internet) auf ein sicheres Netz (zB. internes Firmennetz) erfolgt. Dabei ist das interne Netz meist durch Sicherheitsmechanismen (zB. Firewall) vor unauthorisierten Zugriffen geschützt. Für den Aufbau zur Emulation von Angriffen wurde dieses Szenario nachgebildet. Dabei werden, abhängig von der Intention, die hinter der Emulation steckt, Änderungen an den Einstellungen der Firewall vorgenommen. So kann die Firewall an sich getestestet werden oder aber man lässt eplizit gefährlichen Netzverkehr durch, um die Auswirkungen von Angriffen auf das interne Netz zu testen. Ein weiterer wichtiger Aspekt ist der Einsatz einer Kontrollinstanz, diese überwacht den kompletten Netzverkehr und dient zur Verifizierung der durchgeführetren Emulationen.

2.4 Praxisbeispiel

2.4.1 Aufbau

Um die theorietischen Kenntnisse über die Emulation in der Praxis anzuwenden, haben wir im internen Netzt des Labors für Verteilte Syteme und Internetsicherheit ein Szenatio zum Emulieren von Angriffen nachgebildet.
Dazu wurde zunächst ein hostbasiertes Intusion Detection System (hier Snort) installiert und an die Umgebung angepasst und dementsprechend konfiguriert. Nach der abgeschlossenen Konfiguration ging es nun daran die Funktionswesie von Snort zu testen. Dazu wurde zur Verifikation des Netzwerkverkehrs zunächst ein Paketsniffer parallel zum Intrusion Detection System(IDS) installiert.
Nun mussten potentielle Angreifer im Netz platziert werden, dabei erwies sich die Knoppix-Security Tools Distribution als nützliches Tool. Knoppix STD ist ein vorkonfiguriertes Linux-BS, was von einer bootfähigen CD sofort einsatzbereit ist. So konnten drei(??) Rechner ohne großen Aufwand zu potentiellen „Angreifern“ konfiguriert werden. Als Tool zur Durchführung der Angriffe wurde der Security-Sannner „Nessus“ eingesetzt. Ein weitere Rechner führt gleichzeitig eine DoS-Angriff auf das hostbasierte IDS durch. So ergibt sich folgender Aufbau der Emulationsumgebung.

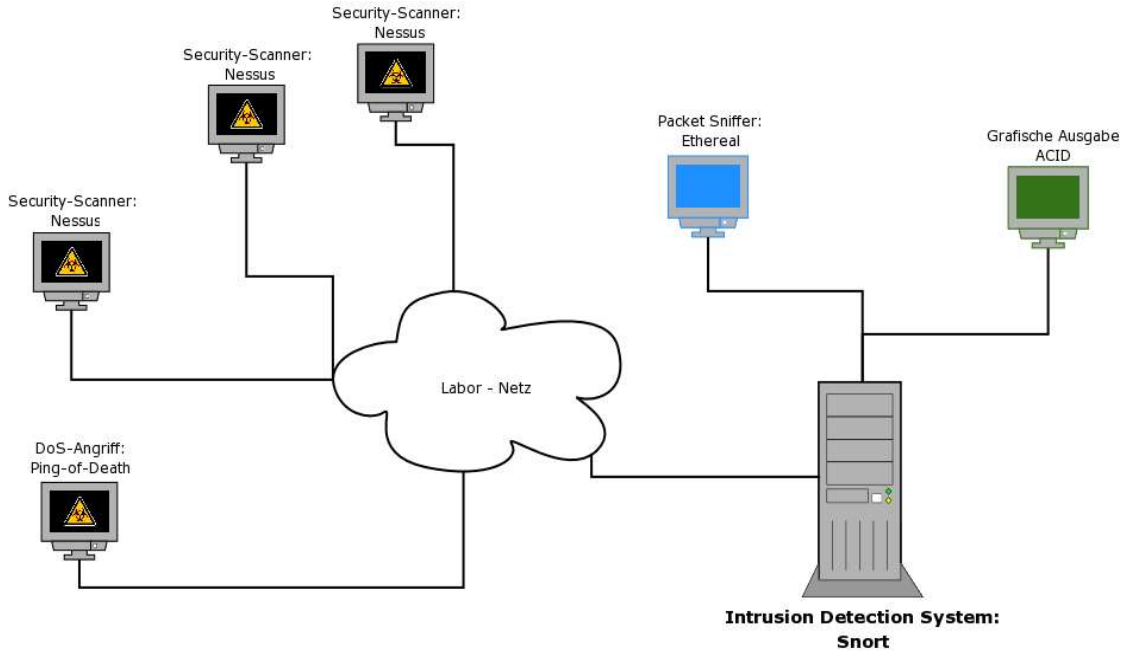


Abbildung 3: Aufbau im Labor

Anhand der durchgeführten Angriffe, die manuell konfiguriert wurden, war wir dann ibn der Lage die funktionsweise des IDS zu testen. Als Kontrollinstanz zwischen durchgeführten Agriffen und erkannten Angriffen beim IDS diente der oben beschriebene Paketsniffer. Die Ergebnis des Snort-Systems zeigt folgendes Bild.

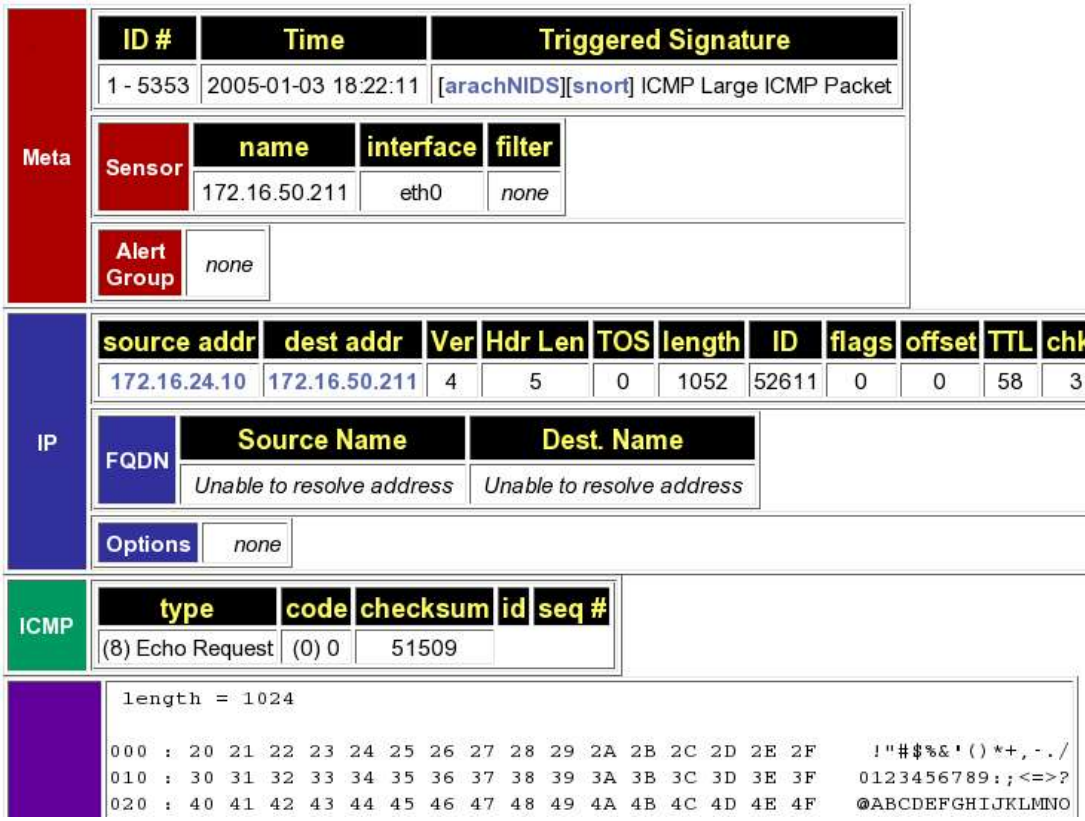


Abbildung 4: Ergebnisse von Snort

Durch den hier beschriebenen Aufbau einer Umgebung zur Emulation von Angriffen war es möglich Angriffe gezielt unf manuell konfigurierbar durchzuführen. So konnte eine sicherheitskritische Anwendung, hier ein Intrusion Detection System, vor dem wirklichen Einsatz auf Schwachstellen und Misskonfigurationen gestestet.

2.4.2 Sicherheitsanwendungen

2.5 Honeypot

Ein Honeypot (deutsche Übersetzung: *Honigtopf*) ist ein Programm (oder ein kompletter Server), das die Aufgabe hat, Angriffe in einem Netzwerk auf sich zu ziehen und Aktionen des Angreifers zu protokollieren.
Die Grundidee ist, in einem Computernetz einen oder mehrere Honeypots zu installieren, die dem legitimen Netznutzer unbekannt sind, und daher niemals angesprochen werden. Ein Angreifer, der nicht zwischen echten Servern/Programmen und Honeypots unterscheiden kann und routinemäßig alle Netzkomponenenten auf Schwachstellen untersucht, wird früher oder später in die Falle tappen. Die bloße Tatsache, dass irgendjemand versucht, mit einem Honeypot zu kommunizieren, wird als potentieller Angriff betrachtet.
Mit Hilfe eines Honeypot-Programmes werden Netzwerkdienste (Mail-Server, Datei-Server, ...) eines einzelnen Rechners oder sogar ein vollständiges Netzwerk simuliert. Erfolgt ein unberechtigter Zugriff auf einen derartigen virtuellen Dienst, werden alle ausgeführten Aktionen protokolliert und gegebenenfalls ein Alarm ausgelöst.
Ein Beispiel für ein Honeypot-Programm ist honeyd.

2.6 Intrusion Detection System

Ein Intrusion Detection System (IDS) ist ein Programm, das der Erkennung von Angriffen auf ein Computersystem oder Computernetz dient. Richtig eingesetzt, ergänzen sich eine Firewall und ein IDS und erhöhen so die Sicherheit von Netzwerken. Man unterscheidet netzwerkbasierte (NIDS) und hostbasierte Intrusion Detection Systeme (HIDS).
Grundsätzlich gibt es zwei Verfahren zur Einbruchserkennung: den Vergleich mit bekannten Angriffssignaturen und die sogenannte statistische Analyse. Die meisten IDS arbeiten mit Filtern und Signaturen, die spezifische Angriffsmuster beschreiben. Der Nachteil dieses Vorgehens ist, dass nur bereits bekannte Angriffe erkannt werden können und durch das Modifizieren bekannter Angriffe ist das System leicht umgehbar.
Andere IDS verwenden heuristische Methoden um auch bisher unbekannte Angriffe zu erkennen. Ziel ist, nicht nur bereits bekannte Angriffe, sondern auch ähnliche Angriffe oder ein Abweichen von einem Normalzustand zu erkennen.

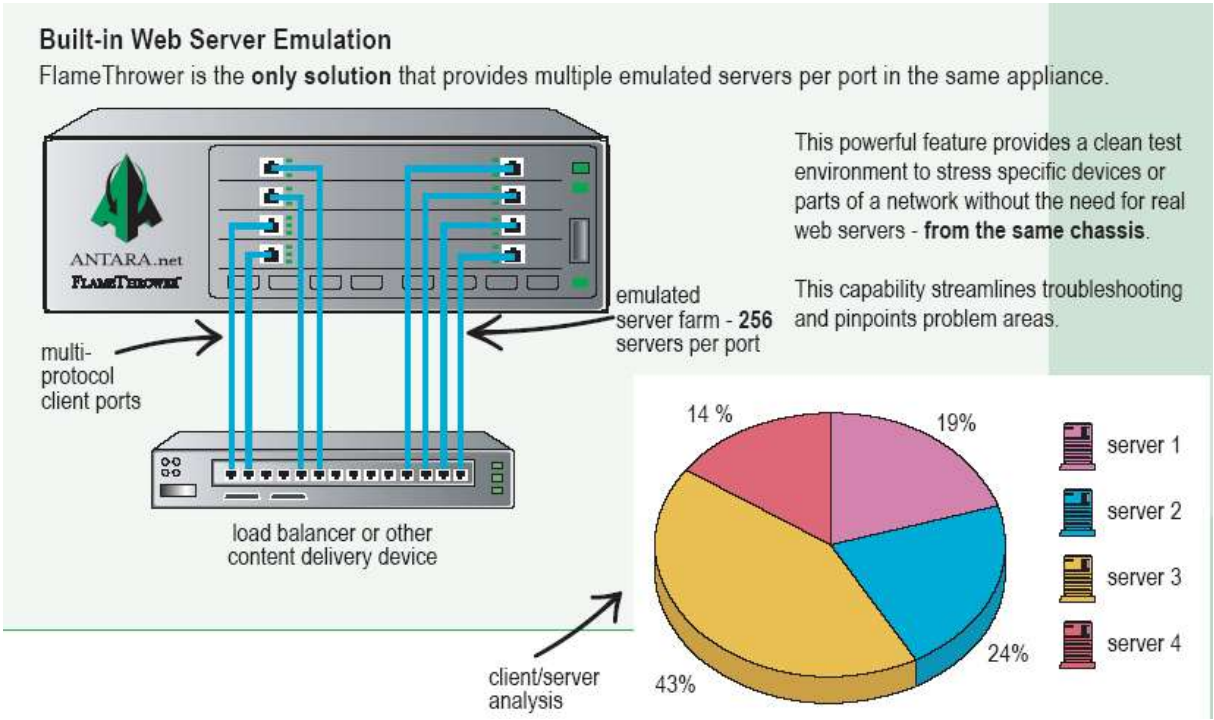
2.7 Firewall Systeme

Firewall-Systeme wurden in der Vorlesung ausführlich behandelt.

2.8 FlameThrower

Abbildung 5: FlameThrower von Antara

Der FlameThrower ist ein hard- und softwarebasiertes Testtool, welches dem Benutzer die Möglichkeit gibt die Netzwerkinfrastruktur, die Abrufbarkeit von Applikationen und die Sicherheit zu überprüfen und zu bewerten. Durch das Emulieren von Internetverkehr kann die gesamte Netzwerkinfrastruktur unter enorme Last gesetzt werden und so erlaubt es der FT Schwachstelle, Mängel und potentielle Flaschenhälse in Netzwerken zu finden bevor es zum Zusammenbruch kommt.



2.8.1 Load Balancer

Der Load Balancer ist ein Lastverteiler, der die Antwortzeiten und Auslastung einzelner Server beurteilen kann und eine Anfrage von außen mit der bestmöglichen Server-Performance bedienen kann.

Features:

- Alle Auswertungen sind frei konfigurierbar, es existiert bereits eine Vielzahl von vordefinierten Auswertungen
- Frei konfigurierbare Ports (Client / Server)
- Schnittstelle zum Datentransfer in MS Excel.

Testabläufe können automatisiert werden, so dass komplette Testszenarien abgefahren werden können.Mit dem FT besteht die Möglichkeit, die Netzauslastung mit, bis zu 2 Millionen emulierten http-Browsern, 32,000 FTP und LDAP Benutzerlogins und file downloads u.v.m. zu testen.

2.8.2 Traffic-Emulation

Es können:

- HTTP 1.0, 1.1
- FTP
- POP3
- TCP

Unterstützte Protokolle:

- TCP Client / TCP Server
- FTP Client / FTP Server
- SMTP Client / SMTP Server
- SMTP SPAM Server
- POP3 Client / POP3 Server
- HTTP Client / HTTP Server
- Burst TCP Client / Burst TCP Server

Für unseren Vortrag von entscheidender Bedeutung ist die Möglichkeit Netzattacken zu emulieren. FlameThrower hilft das Risiko, welches mit Angriffen verbunden ist, zu verstehen. FT verbindet Verkehrsgenerierung, Attackengenerierung und Attackenerkennung in der einen Box. Wird ein System angegriffen ist es ausschlaggebend zu wissen, wie die verschiedenen Sicherheitsprodukte, der Netz-, Mail- und FTP-Verkehr sich verhält und reagiert. Wenn die Sicherheitsmaßnahmen am Ende sind, sollte man wissen wie „load balancers“, web server usw. sich verhalten, ob sie runterfahren und normal wieder hochkommen oder ob Pakete verloren gehen. Da FT die OSI - Schichten 2 bis 7 kennt, kann man den gesamten Verkehr vor und hinter den Sicherheitsmechanismen erkennen.

Mögliche Attacken, die mit dem FT emuliert werden können sind :

- Ping Attack / Ping Reply Detect / Ping Request Detect
- Syn Attack / Syn Detect
- Land Attack / Land Detect
- Smurf Attacks / Smurf Detects
- Unreachable Host Attack
- Unreachable TCP Attack
- Unreachable UDP Attack
- ARP Attack
- ARP Reply Detect
- ARP Request Detect
- Tear Drop Attack
- etc.

Ping-Attacks

Ping-Attacks sind sehr wirkungsvolle Angriffe, weil bei dem Angegriffenen die Bautrate enorm gesenkt wird. Ping-Attacks funktionieren bei jedem Rechner ! Der Angreifer tippt z.B: folgendes ein:

```
ping -f 192.168.0.101
```

Wirkung:
Durch diesen einfachen Befehl werden so schnell wie möglich Pakete an die gewählte IP geschickt. Aber die gewählte IP kann die Pakete nicht so schnell verarbeiten, als wie sie ankommen, deswegen wird die Rechenzeit und die Bautrate enorm gesenkt.

Schutz:
Schützen kann man vor Ping-Attacks nicht 100%, es sei den der Server unterstützt so was. Aber man kann die Pakete ablehnen, dazu installiert man ein PingFlood-Programm oder aktiviert es bei den meisten Programmen z.B.: bei MIAMI steht für registrierten User so ein Funktion zur Verfügung.

Win95 Ping-Attacks
Beim Win95 Ping-Attack benützt man Ping-Attacks aber gezielt auf den Port 65510. Der Port ist einer von vielen Fehler von Win95 bzw. Windoof95. Wird eine Ping Attack darauf ausgeführt, stürzt der Rechner ab. Diese geschieht mit folgendem Befehl:

```
ping -l 65510 address.to.the.maschine
```

Schutz vor Win95 Ping-Attack:
Man kann sich schützen, indem man eine Firewall aufbaut, die gezielt den Port 65510 überwacht.

SYN-Attack

Der Sender schickt ein leeres Datenpaket an den Empfänger, dabei ist das TCP-Flag SYN gesetzt - damit drückt er den Wunsch nach dem Aufbau einer Verbindung aus; nur das erste Datenpaket einer Verbindung hat das SYN-Bit gesetzt. Der Empfänger antwortet mit einem leeren Datenpaket, darin sind die Flags SYN und ACK gesetzt. Der Sender wiederum quittiert den Empfang mit einem leeren ACK-Paket. Nun steht die Verbindung und der Austausch der Nutzdaten kann beginnen.

Dieses Verfahren kann aber auch für einen DoS-Angriff missbraucht werden. Dazu sendet der Angreifer SYN-Pakete mit gefälschter Absenderadresse an das Opfer. Dieser trägt die aufzubauende Verbindung in eine interne Tabelle ein und antwortet mit SYN/ACK.

Allerdings geht diese Antwort an eine gefälschte Adresse, also eine Maschine, die von diesem Vorgang gar nichts weiß und daher die unmotiviert eintreffenden SYN/ACK-Pakete wegwirft. Die Opfer-Maschine wartet und wartet und wartet auf die letzte ACK-Bestätigung, die jedoch nicht kommen kann und wird. Es wird so lange gewartet, bis nach Ablauf eines Timeouts aufgegeben wird - erst dann wird auch der Eintrag in die Tabelle der Verbindungen wieder gelöscht.

Der Angreifer sendet nun in schneller Folge derartig gefälschte SYN-Pakete an sein Opfer, so dass er die internte Tabelle über die bestehenden Verbindungen schneller füllt, als die unsinnigen Einträge durch den Timeout wieder herausfallen. Ist die Tabelle voll, kann die Opfermaschine keine weiteren Verbindungen mehr aufbauen, auch keine legitimen. Das Ziel des DoS-Angriffs ist erreicht.

Die gängige Methode zur Abwehr dieses Angriffstyps sind SYN-Cookies, wiewohl es natürlich auch möglich wäre, auf einem Paketfilter die Rate der zugelassenen SYN-Pakete pro Sekunde zu limitieren.

Land Attack

Sendet SYN-Pakete mit denselben Quell- und Ziel-IP-Adressen an einen Hostcomputer. Hierdurch entsteht der Eindruck, als würde der Hostcomputer Pakete an sich selbst senden. Während der Hostcomputer versucht, sich selbst zu antworten, wird die Ausführung von Windows NT langsamer.

3 Fazit

Die Emulation Angriffen mit oben beschriebenen Open-Source-Anwendungen ist ein probates Mittel seine Sicherheitseinrichtiungen im Testbetreib zu kompromitieren. Bei vorhandener und ,für diese Zwecke, einsetzbarer Infrastruktur, ist der Aufwand zur Durchführung der Emultion angemessen und vertretbar. Anders sieht es aus, wenn eine passende Infrastruktur nicht zur Verfügung steht. Hier ist der Aufwand, eigens für die Emulatiuon eine geeignete Infrastruktur aufzubauen, sehr hoch. Die Ergebnisse einer so durchgeführten Angriffsemulation ergeben validierbare Aussagen über das sicherheitslevel der getesteten sicherheitsanwendungen.

Der FlameThrower verbindet Verkehrsgenerierung, Attackengenerierung und Attackenerkennung in einer Box. Mit diesem Tool, der Firma Antara, ist es möglich real-world-traffic zu generieren ohne die Infrastruktur bereitstellen zu müssen. Will man die Netzinfrastruktur, die Abrufbarkeit von Applikationen und die Sicherheit eines Netzes auf einem hohen Niveau testen, ist der FlameThrower eine „all-in-one“ - Lösung. Beachtet man, dass der FlameThrower kein open-source Produkt ist, muss man sich das Verhältnis Kosten und Nutzen vor Augen führen, da der einmalig zu zahlende Anschaffungspreis im vierstelligen Bereich liegt.

4 Literatur & Links

•Nessus: http:// www.nessus.org

•Snort: http:// www.snort.org

•Knoppix: http:// www.knoppix-std.org

•Packet Sniffer: http:// www.ethereal.com

•Honeypot: http:// www.honeyd.org

•FlameThrower: http://www.antara.net

•Weitere Tools: http://www.computec.ch/software