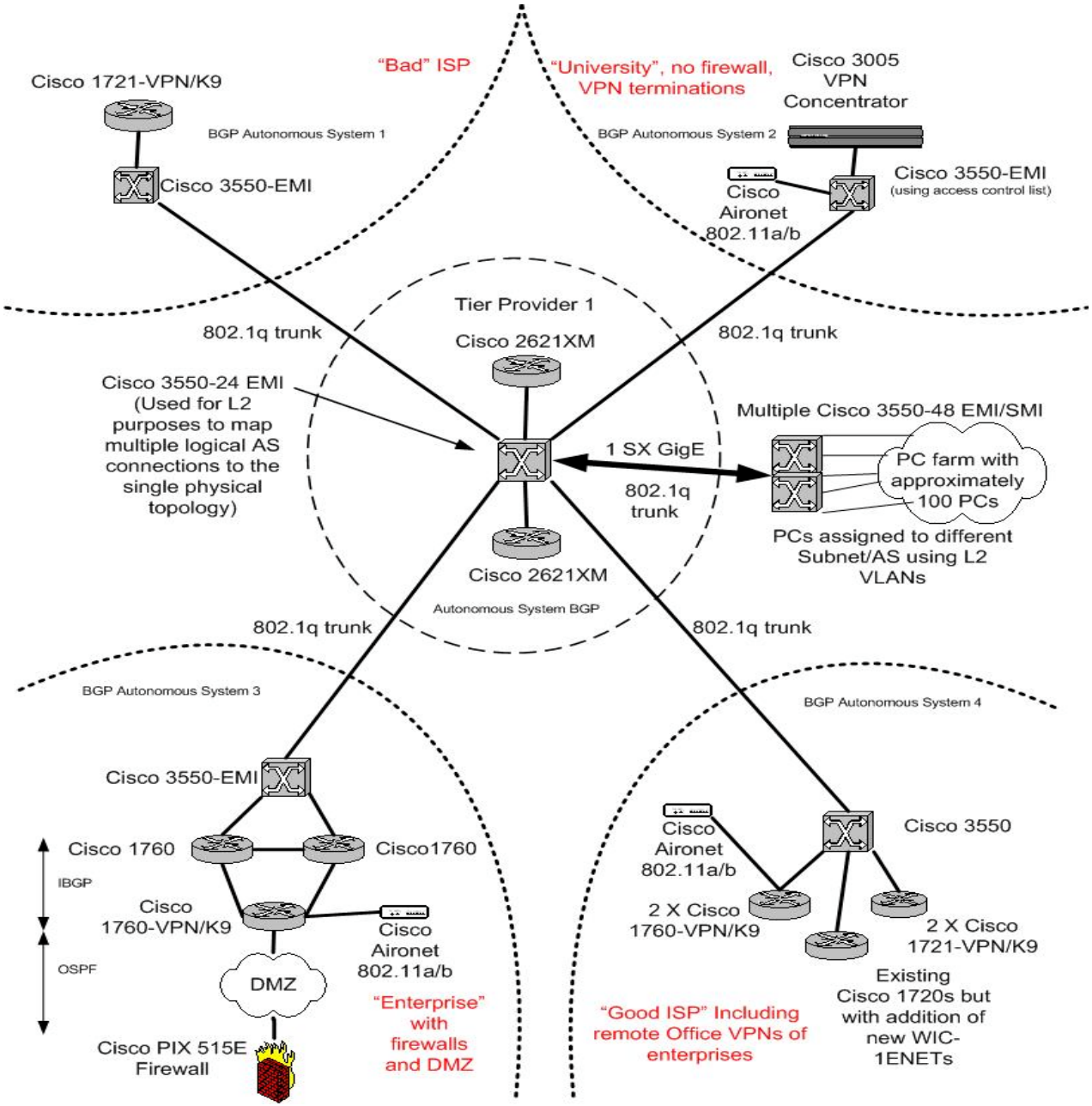
Intrusion Detection Testing and Benchmarking Methodologies

Nicholas Athanasiades, Randal Abler, John Levine,
Henry Owen, and George Riley
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250 USA

Agenda

- Introduction
- Existing Tools and Methodologies
- Examples of Intrusion Detection Evaluation Environments
- Proposed New Environment
- Conclusions


"Bad" ISP
"University", no firewall, VPN terminations
Cisco 3005 VPN Concentrator
Cisco 1721-VPN/K9
BGP Autonomous System 1
BGP Autonomous System 2
Cisco 3550-EMI
Cisco 3550-EMI (using access control list)
Cisco Aironet 802.11a/b
Tier Provider 1
802.1q trunk
Cisco 2621XM
802.1q trunk
Cisco 3550-24 EMI (Used for L2 purposes to map multiple logical AS connections to the single physical topology)
Multiple Cisco 3550-48 EMI/SMI
1 SX GigE
802.1q trunk
PC farm with approximately 100 PCs
PCs assigned to different Subnet/AS using L2 VLANs
Cisco 2621XM
Autonomous System BGP
802.1q trunk
802.1q trunk
BGP Autonomous System 3
BGP Autonomous System 4
Cisco 3550-EMI
Cisco 1760
Cisco1760
IBGP
Cisco 1760-VPN/K9
Cisco Aironet 802.11a/b
OSPF
DMZ
"Enterprise" with firewalls and DMZ
Cisco PIX 515E Firewall
Cisco Aironet 802.11a/b
Cisco 3550
2 X Cisco 1760-VPN/K9
2 X Cisco 1721-VPN/K9
Existing Cisco 1720s but with addition of new WIC-1ENETs
"Good ISP" Including remote Office VPNs of enterprises

Introduction

- Historically, DARPA Intrusion Detection Evaluation program was very significant
- Lincoln Adaptable Real-Time Information Assurance Test-bed (LARIAT) follow-on
- No available common testing environment
- Create small test network and generate attacks or “tcpreplay” captured traffic

Introduction

- What kind of traffic should be used?
- How is realistic background traffic generated?
- Present approaches to testing are inadequate
- Common framework and approach needed!

Introduction

- According to Debar and Morin in "Evaluation of the Diagnostic Capabilities of Commercial Intrusion Detection Systems" RAID 2002: Using 4 of the 5 commercial leader's IDS products in a test, "none of them would be satisfactory"

Introduction

- According to Newman, Snyder, and Thayer in a Network World 6/24/02 review of IDS, “Eight IDSs fail to impress during the month long test on a production network”.
- Incapable Intrusion Detection Systems are a direct result of poor test methodologies during algorithm invention and follow through with product development and evaluation!

Agenda

- Introduction
- Existing Tools and Methodologies
- Examples of Intrusion Detection Evaluation Environments
- Proposed New Environment
- Conclusions

Existing Tools and Testing Methodologies

- Create an individualized and customized test environment
- DARPA IDSs evaluation
- Lincoln Adaptive Real-time Information Assurance Test-bed (LARIAT)
- NIDSBENCH and IDS Wakeup
- Flame Thrower, WebAvalance/WebReflector, Fragrouter, HPing2, Iperf

Existing Tools and Testing Methodologies: DARPA

- 1998/1999 represent the first significant systematic effort to test intrusion detection systems
- Off-Line: 7 weeks of training data available to tune your system under evaluation
- On-Line: 2 weeks of “tcpreplay” data

Attacks in the 1998 DARPA evaluation

	Solaris	SunOs	Linux
Denial of Service (11 types, 43 instances	Back,Neptune, Ping of death, Smurf, syslog, Land, apache2, Mailbomb, Process table, UDP storm	Back,Neptune, Ping of death, Smurf, Land, apache2, Mailbomb, Process table, UDP storm	Back,Neptune, Ping of death, Smurf, teardrop, Land, apache2, Mailbomb, Process table, UDP storm
Remote to Local (14 Types, 17 Instances)	Dictionary, ftp-write, guest, phf, http tunnel, xlock,xsnoop	Dictionary, ftp-write, guest, phf, http tunnel, xlock,xsnoop	Dictionary, ftp-write, guest, imap, phf, named, http tunnel, sendmail, xlock,xsnoop
User to Root (7 types, 38 Instances)	Eject, ffbconfig, Fdformat,ps	Loadmodule, ps	Perl,xterm
Surveillance/ Probe (6 types, 22 Instances)	Eject, nmap, Port sweep, Satan, mscan, saint	Eject, nmap, Port sweep, Satan, mscan, saint	Eject, nmap, Port sweep, Satan, mscan, saint

Existing Tools and Testing Methodologies: DARPA

- Environment for passive intrusion detection systems
- System under evaluation could not query network equipment and respond
- Recorded information was not enough info for some intrusion detection systems

Existing Tools and Testing Methodologies: DARPA

- Made significant contributions toward identifying the complexities and difficulties in testing and evaluating IDSs
- Information assurance community does not recommend using this same approach and data today
- There is no real public domain follow-on effort

Existing Tools and Testing Methodologies: LARIAT

- Goal: “provide tools to assist in the evaluation and configuration of information assurance technologies”
- Not publicly available
- Excellent reference for ideas and methodologies

Existing Tools and Testing Methodologies: LARIAT

- LARIAT emulates the network traffic from a small organization connected to the internet
- User selects profiles for background traffic and attacks
- Traffic and attack scripts are generated
- In 2001 there were 50 attacks available for 9 different operating systems

Existing Tools and Testing Methodologies: LARIAT

- Modified Linux kernel uses one linux machine to emulate many hosts on the network
- Traffic used similar to DARPA 1999 evaluations
- Very powerful methodology but is not widely available

Existing Tools and Testing Methodologies: NIDSBENCH/IDS Wakeup

- Network Intrusion Detection System Test Suite not updated since 1999
- Components include:
- tcpreplay
- idtest – attempts exploits
- fragrouter
- hping- send arbitrary packets
- iwu – sends a buffer as a datagram

Existing Tools and Testing Methodologies: Others

- Other common tools include:
- Flamethrower – commercial load stress tool HTTP/HTTPS
- Web Avalanche/WebReflector – commercial HTTP/SSL/RTP/FTP
- Iperf – used as background traffic source

Agenda

- Introduction
- Existing Tools and Methodologies
- Examples of Intrusion Detection Evaluation Environments
- Proposed New Environment
- Conclusions

Examples of IDS Evaluation Environments: DARPA "like"

- Used to examine Denial of Service detection and response system
- Traffic generated using same technique as DARPA 1998 test bed
- Used attack injection programs
- Reference programs counted number of hung connections on victim server
- Used 10, 15, 30, 40, 60 attacking hosts

Examples of IDS Evaluation Environments: Custom Software

- Evaluation of Intrusion Detection System Appliance
- Used Expect and Tool Command Language Distributed Programming (TCL-DP)
- Expect created Telnet and FTP “users”

Examples of IDS Evaluation Environments:Advanced Security Audit Trail Analysis on Unix

- Wanted to assess the reliability of ASAX
- Tests done in an actual production network!
- Used Trojan Horse executable files, attempted break-ins, masquerading, connections from black listed addresses, privilege abuse, etc.

Examples of Environments:Vendor Independent Testing Lab

- Independent network and security testing facility (NSS Group)
- Used a dedicated test network
- Attack Recognition by running exploits and scans
- Performance under load – stress tests
- IDS evasion techniques
- Stateful operation tests using stick and snot
- Host Performance–CPU utilizations

Examples of IDS Evaluation Environments:Trade Magazine Evaluation

- Trade magazine evaluation of seven intrusion detection systems
- Used a production ISP network
- Normal traffic used as background
- Used old unpatched systems, waited for attacks
- Machines were compromised “immediately”

Agenda

- Introduction
- Existing Tools and Methodologies
- Examples of Intrusion Detection Evaluation Environments
- Proposed New Environment
- Conclusions

Proposed New Environment

- Clear present approaches to intrusion detection test and evaluation are inadequate
- Prevalent present philosophy is to use live or recorded traffic from the site where system will be deployed
- Valid approach but need for public domain test and evaluation methodology exists

Proposed New Environment

- Need sophisticated tools to allow creation of an emulated network placement
- Once recent approach of this type is THOR: A Tool to test Intrusion Detection Systems by Variation of Attacks
- Automatically launches attacks and collects alarms from systems under test
- Repeatability and common test metrics

Proposed New Environment

- Need realistic background traffic generation methodologies
- Need ability to merge attack traffic with background traffic from previous runs or live runs
- Need ability to generate and respond to traffic in real time
- Should work with real network segments as well as captured network traffic

Proposed New Environment

- Need to work with real network segments as well as captured network segment traffic
- Need background traffic generator ability to parse existing traffic traces and merge additional background or attack traffic
- Need real time since IDS may adapt or provide feedback
- Realistic user models and network activity required

Proposed New Environment

- Attacks should be maintained in an attack repository
- Ability to incorporate existing standard attack and evaluation tools
- Automated method for launching and scoring attacks needed

Conclusions

- Tools and Methodologies for uniform testing of intrusion detection systems do not yet exists in the public domain
- Not possible at present to perform comparative tests and evaluations in other than subsets of anticipated deployment environments
- Result of a new methodology having proposed capabilities would be a more uniform test and evaluation capability