CORE 01
CanSecWest 01

Présentation OSSIR

Mardi 12 juin 01

Observatoire
de la Sécurité
des Systèmes
d'Information
& des Réseaux

Franck.Veysset@intranode.com

SOMMAIRE

Franck.Veysset@intranode.com

1. Introduction

Ce document est un compte rendu de la conférence CanSecWest, qui s'est déroulée les 28, 29 et 30 mars 2001 à Vancouver, Canada.

Il s'agit de la deuxième conférence de ce type, organisée à Vancouver.
La principale caractéristique de cette conférence est de réunir ce qui ce fait de mieux en terme de sécurité informatique en un seul lieu. Le contenu des conférences est mitigé, entre un contenu traditionnel et bien pensant, de type « White Hat », et un contenu beaucoup plus underground, qualifiable de « hacker », appelé « Black Hat ».

Le public de ce type de conférence est très varié, mais plutôt de type « underground ».

Il y avait environ 120 à 140 personnes d'inscrite à cette conférence. 50% provenant des USA, 25% de la proche région de Vancouver. Au niveau européen :

- 4 Français (Dont 2 personnes d'Alcatel)
- 2 Suisses (en fait des français de COLT Télécom / securite.org)
- Quelques Anglais (3 ou 4 ?)
- Quelques personnes d'Amsterdam (3~4)
- Quelques personnes d'Allemagne (3~4)
- Quelques asiatiques.

2. Présentation des acteurs

Ci dessous une rapide présentation des acteurs de cette conférence, ainsi que l'objet de leur conférence. (en anglais…)

Renaud Deraison - Author of Nessus, speaking about the Nessus attack scanner, giving an overview of scanner operations and a tutorial on Nessus Attack Scripting Language (NASL).

Martin Roesch - Author of the popular Snort Intrusion Detection System (IDS), speaking about new developments in IDSes.

Dug Song of Arbor Networks - Author of many famous networking tools. Speaking about monkey in the middle attacks on encrypted protocols such as SSH and SSL. :-)

Rain Forest Puppy - Will be speaking about assessing the web, with demonstrations of several new (previously unreleased) rfp.labs web tools including the release of RFP Proxy and other surprises in his inimitable style.

Mixter of 2XS - Author of several widely used distributed tools and some popular security whitepapers will give a talk about "The future of distributed applications" explaining the key elements of peer-to-peer networks, discussing a few examples/possibilities of distributed technology, and related security problems in distributed networks.

K2 of w00w00 - Will present his new ADMutate, a multi-platform, polymorphic shell-code toolkit and libraries for detection evasion.

Matthew Franz of Cisco - Author of Trinux: A Linux Security Toolkit, will discuss a comprehensive security model (including tools and techniques) for conducting security evaluations of firewalls, VPNs, and other networked devices.

Lance Spitzner of Sun - Will present more of the HoneyNet group's honeypot findings, including watching Romanian hackers on their own web cam while they were hacking one of his honeypots for their botnet.

Theo de Raadt is the principal architect of the OpenBSD operating system project. Will present the OpenBSD project.

Fyodor of insecure.org - Author of the popular nmap network scanner, will talk about new mapping and scanning tools and techniques.

HD Moore of Digital Defense - Will give a surely popular talk about his more esoteric NT/Win2k penetration test tricks in a presentation called "Making NT Bleed." where he will cover some of the procedures he has had to develop during the course of cracking multiple systems for customers daily.

Jay Beale of MandrakeSoft - Author the Linux Bastille scripts and Security Team Director at MandrakeSoft, will talk about securing Linux.

Kurt Seifried of SecurityPortal.com - Will talk about cryptography a "two edged sword" including PKI, SSH and SSL.

Dave Dittrich of The University of Washington - Author of many famous Forensic Analyses and UW Senior Security Engineer, will give a talk about finding intruders, then tracing their actions through the trails they leave on penetrated systems.

Robert Graham of NetworkICE - CTO of NetworkICE, will discuss IDS operations and decoding technology, illustrating with exploits including his new "sidestep" utility during live demonstrations of the BlackICE Sentry IDS system and other IDSes like Snort.

Sebastien Lacoste-Seris & Nicolas Fischbach of COLT Telecom AG - Editors of the French Securite.Org site, will discuss the rollout of Kerberos across their company and hosting center using Kerberized SSH and Kerberos V5 across Unix/Cisco/Win2k platforms to provide

strong authentication with SSO capabilities, their experiences, and what potential problems and limitations they faced.

Andrew Reiter, R&D Engineer with Foundstone, Inc., is an experienced computer and network security researcher with a great interest in discovering new methodologies for exploitation and protection of systems, reverse engineering, protocol development, and FreeBSD kernel programming. He has worked in doing security research and development with numerous groups including the BindView RAZOR team and multiple well-known, non-profit security research groups. Andrew has developed new talks and presented on numerous occasions for New Dimensions, Inc., where he has spoke on hacking and security to major government and military personnel, and also has spoke in front of various technical groups at meetings and conferences on topics ranging from FreeBSD kernel development to security. Andrew is currently working at Foundstone, Inc. where he is part of a security research and development team.

Christopher Abad, an R&D Engineer with Foundstone, Inc., is currently studying mathematics at UCLA and has also done considerable research in the security industry including pioneering work in the concepts of passive network mapping. He has given various presentations on this subject at security conferences including Defcon.

The will talk on “Win32 Format String Exploits”.

Gary Golomb has been working with Dragon for some time now. He has recently done some work with Greg Hougland where he did a rather thourough analysis of the NT Rootkit. Gary would like to present a talk along the lines of "Stateless TCP connectections and their effect on network IDS". It turns out that tools like NT Rootkit and a lot of the DDOS clients don't need a full three way handshake to establish a session. This mucks with NIDS to no end.

Andrew R. Baker is a senior software engineer at farm9. He will be talking about advanced Snort techniques.

L’organisateur de cette conférence était Dragos RUIU (DURSEC.COM, Founder).

Elle se déroulait dans un hôtel de Vancouver. (à noter : Chambres câblées Internet, couverture Wavelan 802.11b de la salle de conférence).

Le site web de la conférence est :

http://www.cansecwest.com

3. Contenu des présentations

Les présentations se sont déroulées du mercredi 28 mars 12 heures, au vendredi 30, 18 heures.

3.1. Mercredi 28 mars

3.1.1. securite.org, "Kerberos in an ISP environment".

Présentation par Sébastien Lacoste-Seris & Nicolas Fischbach, Colt Telecom.

Agenda :

- Présentation de l'intérêt du protocole Kerberos
- Explication sur le fonctionnement de Kerberos
- Problématique de l'utilisation
- Déploiement en environnement Unix, Cisco et Win2K.

Les transparents sont disponibles sur :
http://www.securite.org/presentations/krb5/

Quelques problèmes pratiques :

Lors de l'utilisation de Kerberos avec des équipements faisant de la translation d'adresse (NAT), les tickets sont rejetés, car l'adresse de la machine est codée en dur dans le ticket. Il est donc nécessaire d'indiquer, lors de la génération du ticket, l'adresse de la machine réelle (donc après la NAT). Le problème se pose lors de l'utilisation de plage dynamique de translation d'adresse, car alors, il faudrait pouvoir indiquer toutes les adresses possibles que peut prendre la machine. (pas de notion de classe ou de mask d'adresse dans le ticket…)

Attaques contre Kerberos :

- Vulnerability in Kerberos password authentication wia KDC AS spoofing
(http://www.monkey.org/~dugsong/kdcspoof.tar.gz)
- Replay attacks:detected (C+S are time synchronised)
    - exposed keys: keys have a limited lifetime but are multisession keys
    - temporary file vulnerability.
- Password guessing: use a good passphrase
- Trojaned clients: use an OTP
- Implicit trust between realms (ticket forwarding...)
- The KDC must be completely secured; It is very sensible...

Commentaires :

Présentation intéressante, avec un bon retour d'expérience. Il faut noter que COLT utilise Kerberos afin d'avoir une administration centralisée de ces administrateurs, ainsi qu'un pseudo SSO (Single Sign On). L'expérience fonctionne avec un nombre très restreint d'utilisateur.

3.1.2. Renaud Deraison, *The Nessus project*

La présentation est articulée en deux parties :

- Brève introduction sur le scanner Nessus
- Petit tutorial sur l'écriture de scripts en NASL

1 – Le scanner Nessus

Le projet Nessus : environ 3 ans de travail pour en arriver là.

Le produit Nessus a été médiatisé au cours de ces derniers mois. Il a été présenté comme le produit numéro 1 lors du sondage « *best security tools survey* » émis par Fyodor.

Explication de l'architecture de Nessus, du mode client serveur…

Actuellement, Nessus effectue plus de 630 tests.

Renaud explique alors que le produit est devenu ce qu'il est grâce aux soumissions de ces utilisateurs. En effet, la roadmap a été grandement construite grâce aux différentes *wishlist* des utilisateurs. Les retours d'expériences, les remontés de bugs, les donations de matériel ou de code ont permis une bonne croissance du projet Nessus.

2 – NASL

Présentation du langage NASL (*Nessus Attack Scripting Langage*), permettant l'écriture de scripts Nessus. Il faut noter que les scripts Nessus peuvent être écrits en C (historiquement) ou en NASL.

Ce que NASL n'est pas : NASL a été écrit dans le but de l'utiliser pour l'élaboration de plugins. C'est donc un langage très typé pour cette utilisation. NASL est donc beaucoup moins puissant que des langages tels que PERL ou C, mais il est conçu avant tout pour la réalisation de plugins. Des fonctionnalités de sécurité ont donc pu être intégrer, permettant par exemple de rendre pratiquement impossible l'écriture de chevaux de Troie en NASL.

Les plugins NASL fonctionnent dans un bac à sable (model « Sand Box »).

La suite de l'exposé détaille l'écriture d'un script de type "scanner SSH", puis l'amélioration de ce script.

3.1.3. How secure is secure? Matthew Franz, Security Technologies Assessment Team, Cisco

"Conducting threat oriented product security evaluation".

M. Franz est connu dans le monde de la sécurité pour son travail sur Trinux, une distribution orientée auditeur de Linux (fourniture d'un kit d'outil d'audit et d'un Linux opérationnel sur un ensemble réduit de diskettes).

Plan de la présentation :

- Introduction
- A threat model
- A testing methodology
- Sample product eval
    - SOHO firewall / router
- Sample product eval
    - IPSec

Le but de la présentation est d'introduire une méthode de validation/test de produit d'un point de vue sécurité. L'orateur concentre sa présentation sur l'explication des besoins, une définition précise des objectifs et des moyens à engager, afin de déterminer une démarche méthodologique optimale du test de produit.

Cette méthode, une fois présentée, sera appliquée à deux exemples concrets, un boîtier firewall routeur et le protocole IPSec.

Contacter l'auteur :
personal Email: mdfranz@io.com
work mail: mfranz@cisco.com

Le support de cette présentation est disponible à l'adresse :
http://www.io.com/~mdfranz/papers/

3.2. Jeudi 29 mars

3.2.1. Win32 format string Exploits

Conférence sur les "format string attacks".
Par Andrew Reiter et Chris Abad, Ingénieur R&D, Foundstone, Inc

Présentation des concepts d'attaques sur les chaînes de format.
La présentation est en fait articuler sur des explications, puis des exemples concrets d'attaques de type buffer overflow. Le tutorial est complet, détaillant même l'utilisation de débuggeur de type *gdb* ou *softice*.

3.2.2. Digging Through Compromised Systems and Tracking intruders

Présentation de Dave Dittrich dittrich@cac.washington.edu

Cette présentation a pour objet de montrer un ensemble de démarche permettant l'analyse « post-mortem » d'une machine après compromission / intrusion.

Plan de la présentation :

- Intro & background
- Sources of data
- Getting data (gently) off the system
- Analysis
- The forensic Challenge
- Basic steps on forensics Analysis of Unix systems
- Lots of pointers

La démarche adoptée consiste en une analyse du système en plusieurs étapes :

- Analyse externe, par l'utilisation d'outil de type nmap…
- Analyse interne de la machine, par analyse des fichiers journaux, checksum md5…

Il est intéressant de pouvoir extraire les données de la machine compromise sur des supports externes, tels que CD, bandes… Ensuite, l'utilisation d'outils avancés, tel le « Coroner's Toolkit » va permettre d'obtenir un ensemble d'information vital pour l'analyse des évènements intervenus contre cette machine.

L'étapes suivante de l'exposé a consisté en une analyse détaillée du « *forensic challenge* » mis en place dans le cadre du projet « honeynet ». Ce projet consistait en une analyse d'une machine compromise. L'ensemble des éléments concernant cette machine était mis à disposition des candidats (images de disques durs, traces réseaux). Les candidats devaient retrouver les éléments ayant conduits à la compromission de cette machine.

Plusieurs personnes ont répondu à ce projet. Les éléments intéressants sont disponibles sur le web du projet : http://projet.honeynet.org

Quelques chiffres marquants :

- Temps nécessaire pour compromettre le système: approximativement 30 minutes
- Temps moyen d'investigation sur incident : 48 heures
- Coût moyen de diagnostique de cet accident (en contexte commercial)
  - (@us 70k$/yr)) : US$2067 +/- $310
  - Coût estimé pour une intervention de professionnels : (@US$300/hr.): US$22,620

Contacts et information sur cet exposé : http://staff.washington.edu/dittrich

3.2.3. Peer to peer and the future of distributed applications

Présentation effectuée par Mixter.

Cette présentation vise à introduire la notion de peer to peer, un modèle de communication décentralisée, ou plusieurs acteurs jouant le même rôle, communiquent entre eux.

Ce modèle présente de bonnes propriétés d'évolutivité / modularité / redondance, car chaque système est indépendant des autres, et le système global pourra continuer à fonctionner même en cas de problèmes sur un nombre important de ces composants.

Exemple d'applications reposant sur ce modèle : Freenet, Gnutella, hacktivismo…

Des systèmes tels que Napster ou IRC reposent en partie sur ce modèle, mais nécessitent des nœuds centraux d'interconnexion.

La présentation de Mixter a ensuite détaillé les différents problèmes associés à ce type de technologie, tel que les contraintes liées à la confidentialité ou l'anonymat, puis a dressé une vision de l'avenir de ce type de solution, en introduisant le projet **Hacktivismo,** un système de communication sans censure, gratuit et libre sur Internet.

Contacts :

http://mixter.void.ru

http://mixter.warrior2k.com

mail : mixter@2xss.com

3.2.4. Polymorphic Shellcode

Présentation de K2 of w00w00 [http://www.ktwo.ca].

K2 a présenté un nouvel outil, ADMutate, dont le but est de rendre indétectable par les IDS le code typique, généralement facilement identifiable, utilisé pour les attaques de type « débordement de pile ».

Contenu de la présentation :

- NIDS evasion
- Polymorphic properties
- buffer overflows
- Implementations
- Smart features

Un IDS réseau va généralement analyse le trafic, et reconnaître via de l'analyse / reconnaissance de chaînes / analyse syntaxique (*pattern matching*) des signatures d'attaques. Par exemple, les attaques de type « buffer overflow » seront généralement

détectées par reconnaissance d'instructions assembleur « *NOPS* » ou d'instructions shell « */bin/sh* ».

L'approche de K2 est d'utiliser différents codages pour camoufler ces instructions.

Plusieurs méthodes sont possibles :

- Niveau Réseau
    - IP fragmentation
    - Spoofed data
- Niveau applicatif
    - Data obfucation
        - Unicode
        - alternate operator (.. = \)
    - Code/Data encoding
        - polymorphism

L'attaque « anti-ids » est en fait un toolkit permettant de générer des shellcodes polymorphes. L'idée est donc de rendre les attaques à base de buffer overflows beaucoup plus difficiles à détecter en masquant par exemple les séquences de NOP (0x90 sur Intel) ou les chaînes /bin/sh. Celles-ci sont souvent utilisées par les IDS pour détecter une tentative d'exploitation d'un service distant.

Le toolkit se présente sous forme d'une API. On applique des routines de cette API sur un shellcode (type, execve("/bin/sh",..)) pour en générer un nouveau, crypté, qui est ensuite auto décrypté lors de l'exploitation réussie (utilisation de XOR). La technique elle-même est bien connue dans le monde des virus infecteurs d'exécutables. Les plates-formes supportées sont Intel, Sparc, HP (pa), et MIPS.

3.2.5. The honeynet project

Présentation par Lance Spitzner (Sun Expert Security services…)

Cette conférence vise à introduire le projet « Honeynet », ainsi qu'à présenter son fonctionnement et quelques résultats et faits.

Le projet Honeynet est une initiative d'une trentaine de professionnels de la sécurité informatique, destiné à apprendre les outils, tactiques, motivations et méthodes utilisés par la communauté « Blackhat » sur l'Internet. Cette information est alors redistribuée librement, afin d'informer et de sensibiliser les personnes intéressées par la sécurité informatique.

Un « Honeynet » est un réseau informatique, constitué de machines opérationnelles banalisées, destinées à être attaquées, et éventuellement compromises, afin de pouvoir apprendre en analysant les méthodes et/ou éléments ayant conduis à sa compromission.

La démarche du projet Honeynet est très intéressante, car contrairement aux démarches plus préventives, tels que l'analyse de traces IDS, ce projet donne une vision plus active,

avec une approche beaucoup plus pratique permettant d'appréhender le comportement et les actions « blackhat ».

Lance a d'abord introduit l'architecture de la plate-forme Honeynet, puis a ensuite expliqué rapidement deux attaques survenues au cour des derniers mois sur ce réseau :

- Une attaque sur un système Windows 98, via l'utilisation d'un partage mal protégé (configuration par défaut)
- Une attaque sur un système Solaris 2.6, via l'utilisation du démon *rpc.ttdbserv.*

Information et pointeurs :
http://projects.honeynet.org

3.2.6. Monkey in the middle (and other public-key quandaries)

Dug Song est l'auteur très connu de l'outil « Dsniff 2.3 », publié en décembre 2000. Lors de la sortie de cet outil, de nombreux média ont annoncé la fin des technologies de type SSL et SSH, dû aux possibilités de cet outil qui exposait au grand jour plusieurs attaques possibles contre celles-ci.

En effet, DSNIFF permet très facilement de mettre en œuvre des attaques de type « man in the middle », permettant alors l'interception du trafic normalement chiffré.

Dug a d'abord introduis les technologies de type « clé publique », avant de passer en détail plusieurs problèmes associés à ce type de technologie.

Les principaux problèmes sur ce type de technologies concernent essentiellement l'absence d'une gestion centralisée des éléments de sécurité. Ainsi, la multiplicité des autorités de certification (la plupart commerciales), dans le cas de SSL et de HTTPS, ou l'absence d'annuaire centrale (cas de SSH et de PGP) font que l'utilisation de ces technologies présente des risques. Ainsi, aucune authenticité ni unicité ne peuvent être accordées sans homogénéité et coopération sur l'espace de nommage.

Dug a alors pris l'exemple de son université, ou il a montré deux entrées distinctes de l'annuaire de son université, chacune pour un Mr Douglas J. Song… Qui est qui ?

Les détails concrets d'attaques par utilisation de certificats de type « self signed » ont ensuite été exposés.

La non utilisation des CRL, listes de révocation de certificats, rajoute encore un niveau d'insécurité au problème. Certains produits reposant sur de la cryptographie à clé publique n'implémentent même pas ce type de fonctionnalité (tel que Internet Explorer par exemple).

Dug Song a fait référence aux certificats Verisign attribués récemment de façon erronée au nom de Microsoft.

Support :
http://www.monkey.org/~dugsong/talks/cansecwest01/

3.2.7. Why Whisker Sucks – « Challenges of auditing online web applications »

Présentation par Rain Forest Puppy, auteur du scanner de CGI « Whisker ».

Les serveurs Web ont envahi les réseaux informatiques… De plus en plus d'équipements offrent en effet des interfaces Web. Malheureusement, de plus en plus de produits ne sont plus conformes aux RFC, et apportent des modifications propriétaires aux spécifications et normes de HTTP.

Whisker est un outil dédié à l'audit de serveur Web. A la base, il s'agit d'un scanner CGI, disponible aussi sous la forme de librairie, permettant alors de scripter ces fonctions afin de développer ses propres applications.

Whisker dispose aussi de fonctions avancées de type « anti-ids », attaque par brut force des mots de passe…

Néanmoins, Whisker est un produit limité. Les configurations diverses et non-standards des serveurs web ainsi que le recours à des bases de signature pour le fonctionnement font que Whisker est parfois limité dans son fonctionnement.

Une solution élégante et performante à l'audit de serveur Web peut reposer avantageusement sur l'utilisation d'un « proxy » adapté :

- Un proxy est situé à une place parfaite pour la réalisation d'un audit : entre le client et le serveur
- L'auditeur n'a plus de contrainte concernant le navigateur qu'il utilise
- Les connexions sortantes (ou les réponses entrantes) peuvent être filtrées et modifiées à la volé

RFP a ensuite détaillé différents produits capables d'assurer des fonctions de type « proxy orienté audit » ou scanner de CGI, tels que :

- Pudding : sur le model Proxy, info sur http://www.sensepost.com
- The ELZA, présenté sous la forme de bilbiothèque, http://www.stoev.org
- Webinspect, scanner traditionnel, http://www.spidynamics.com
- Achilles, sur le model Proxy, avec support SSL, http://www.digizen-security.org
- Appscan, produit commercial, http://www.sanctuminc.com
- Et bien sur Whisker, http://www.wiretrip.net/rfp

RFP travaille actuellement sur un nouveau produit, RFProxy, qui supportera un ensemble de fonction avancé, orienté audit et modification de données à la volée. Le produit sera disponible très prochainement sur le site Web de RFP.

Fonctionnalité de RFProxy :

- Fonctionne en mode proxy
- Modification des données à la volée
- Journalisation du trafic entrant et sortant
- Modification des entêtes HTTP (et des cookies)
- Possibilités de fonctions anti-ids (fonctions de Whisker V1.4)
- Réécriture HTML (formulaire, refer…)
- Ecrit en perl

RFProxy sera disponible sur :
http://www.wiretrip.net/rfp/cansecwest/

Le support de cette présentation est disponible sur le site de RFP.

3.2.8. Making Windows NT bleed....

Présentation par HD Moore, de Digital Defense Inc.

Pourquoi parler de NT et de ses problèmes de sécurité ?

- Parce que 20% des serveurs Web sur Internet fonctionnent sous NT
- Parce que c'est une plate-forme populaire dans les milieux de la finance
- Parce que c'est un système très difficile à sécuriser
- Et parce que c'est donc un système (relativement) simple à pirater…

Les principaux caractéristiques / problèmes d'IIS sont :

- Très facile à mettre en place
- Installation par défaut très vulnérable
- Les fichiers d'exemples sont particulièrement vulnérables
- Les extensions par défaut ont des problèmes largement connus
- Le fameux bug « UNICODE »
- Les services RDS (Remote Data Services)…

Les vulnérabilités associées à ces différents services sont décrites dans la présentation associée à ce compte rendu.

Suite à cette présentation générique de IIS, HD Moore a ensuite détaillé 4 scénarios d'attaques contre des serveurs IIS :

- En utilisant une installation par défaut d'un serveur Microsoft SQL
- Via l'utilisation de MDAC 1.5
- Via SQL RDS
- Et finalement via le bug UNICODE, en uploadant un serveur VNC

Des informations complémentaires sont disponibles sur :
http://www.digitaldefense.net/csw/
Contact de l'auteur :
hdmoore@digitaldefense.net

3.3. Vendredi 30 mars

3.3.1. Un-TCP and driver-programs – "On breaking Protocols, and Not playing by OS rules"

Gary Golomb, Enterasys Networks / Network Security Wizards

Cet exposé vise à introduire une nouvelle vision sur le comportement de certains « Rootkits » et les nouvelles méthodologies utilisées.

Une étude de nouveaux rootkits a permis de découvrir de nouvelles technologies de camouflage très perfectionnées. La société « Enterasys » travaillant aussi sur des IDS (et notamment DRAGON), des études sur les signatures des outils sont indispensables pour parvenir à les détecter.
La nouvelle génération de rootkit est particulièrement furtive.

Sur le poste local :

- Installation au niveau du « kernel mode » et non dans le « user mode »
- Le rootkit peut alors intercepter les appels bas niveaux, tels que les accès à la base de registre, et se camoufler (supprimer toutes les traces qu'il laisse)

Au niveau des communications distantes :

Plusieurs approches existent :

- Trafic normal : TCP
- Trafic un peu moins classique : UDP
- Trafic étrange : violation des RFC, exemple de Loki et de Rootkit

Exemple de trafic anormal :
Utilisation de TCP dans un mode sans état (Stateless TCP)

- Début d'échange de donnée avant l'établissement complet de la connexion TCP (S / SA / échange de donnés puis A…)

- Encapsulation de trafic dans un flux Telnet, mais… le trafic continu après un échange de paquet « Fin » sensés terminé la connexion.
- Le contrôle de taille de fenêtre TCP n'est plus utilisé

Autres phénomènes étranges : Lors de l'émission d'un paquet RST à la machine compromise pour arrêter la connexion, le rootkit répond avec un paquet RST, et continu ses communications…

Ainsi, la détection de ce type de rootkit peut être très difficile avec des IDS classiques, car le non-respect des règles d'usage de TCP est quelque peu déroutant. Les réponses automatiques des IDS (contre-mesures) sont aussi inefficaces.

Des informations complémentaires sont disponibles sur :
http://www.rootkit.com

3.3.2. IDS evasion vs. protocol-analysis

Robert Graham, CTO of network ICE (high-spedd IDS)

L'orateur de cet exposé a une excellente connaissance des protocoles de l'Internet, car il a travaillé plusieurs années pour des sociétés éditant des outils d'analyse protocolaire (packet sniffer).

Cet exposé vise à présenter quelques techniques anti-ids. Plusieurs technologies existent dans le domaine des IDS :

Analyse des couches basses :

- Décodage IP et ré-assemblage
- Décodage TCP et ré-assemblage
- Processeurs HTTP
- Analyses statistiques

Analyse des couches applicatives :

- Recherche de contenu (payload search)
- Analyse de protocole

La technologie qui nous intéresse aujourd'hui est l'analyse de protocole. Au lieu de faire du « pattern matching », l'analyse de protocoles consiste à parser les données, en essayant de reconstruire les divers niveaux d'encapsulation de donnée pour ne s'intéresser qu'aux données intéressantes. L'analyse se fait alors au niveau des champs spécifiés dans chaque protocole, à la recherche d'anomalies au niveau « applicatif ». La compréhension du protocole permet aussi d'introduire la notion d'état au niveau même du protocole.

L'intérêt de l'analyse protocolaire est que la détection d'événements anormaux ne repose plus uniquement que sur des bases de signatures. Il est donc beaucoup plus difficile de camoufler des attaques. De plus, ce type de technologie IDS est en mesure de détecter un ensemble d'attaques beaucoup plus important, voire même de nouvelles attaques inconnues, par analyses des éléments transmis au niveau du protocole.

L'inconvénient de ce type d'analyse est la complexité apportée au niveau du décodeur. La complexité est toujours dangereuse quand on fait de la sécurité. De plus, la mise à jour du décodeur de protocole est une action lourde, plus complexe qu'un simple update d'une base de règles.

Les performances de ce type d'analyse sont plutôt satisfaisantes, et ne croissent pas en fonction de la taille de la base de signature [O(log(n)) / O(n)]

Un des gros intérêts de ce type de détection est la capacité de détecter des variations au niveau des attaques existantes. Ainsi, quand le décodeur est capable de comprendre un protocole tel que DNS, tous attributs ou champs anormaux dans ce protocole seront alors détectés et remontés. Ceci fonctionne bien, au détail prés que certaines implémentations ne sont pas conformes au RFC…

Présentation de SideStep

Afin de montrer la pertinence de l'analyse protocolaire, Robert Graham a écrit un petit outil de camouflage, « SideStep ».

Le principe est assez simple, et repose sur les même concepts que Whisker et ses attaques « anti-ids ». Le codage des chaînes d'attaques Web au format UNICODE est un bon exemple de camouflage.

Il est généralement possible de changer les chaînes contenant les attaques, sans en modifier la signification.

SideStep est un petit outil supportant les protocoles HTTP, DNS, SNMP, FTP, RPC et BackOrifice. Il permet de modifier le format des requêtes de façon à les rendre indétectables si l'IDS utilise des techniques à base de fichier de signature.

Exemple, encodage de : dig -t txt -c chaos version.bind@target

Plus d'informations :

http://www.robertgraham.com

http://ww.networkice.com

mail : core1@robertgraham.com

3.3.3. Introduction to Bastille Linux

Jay Beale, Lead developer, Bastille Linux (Security Team, MadrakeSoft).

Présentation des travaux en cours sur Bastille Linux, un ensemble de script pour la sécurisation des distributions Linux RedHat 7.0 et Mandrake 8.0.

(Des travaux sont en cours pour un portage de Bastille sous Slackware et IRIX…)

Que fait Bastille :

Bastille commence par mettre en place un firewall, afin de renforcer la sécurité réseau de la machine. Les démons et programmes inutiles sont ensuite désactivés. Enfin, le système est reconfiguré afin de monter le niveau de sécurité. Plusieurs droits d'accès sont modifiés, il y a par exemple suppression de certains bits Setuid, ou configuration de certains services réseau afin qu'ils soient chrooté et qu'ils ne fonctionnent plus sous Root.

Bastille est un outil très utile, car beaucoup de distribution Linux ne sont pas sécurisée par défaut. Un point important, comme dans tous les travaux touchant la sécurité, est l'éducation des utilisateurs / administrateurs. La sécurité est l'affaire de tous, et tout le monde prend des risques lorsque l'on connecte une machine à l'Internet.

Historique :

- 1.0 : scripts basics de sécurité
- 1.1 : fonctionne sur des systèmes non vierges. Fonctionne en graphique console (Curses). Fourniture d'API.
- 1.2.0 : disponible très prochainement. Configuration sous X-Windows.

Environ 50000 personnes utilisent Bastille Linux (estimation)

3.3.4. Stealth Scanning & IDS Evasion

Présentation par Fyodor, l'auteur (célèbre) de NMAP.

Fyodor introduit dans cette présentation diverses méthodes de scan de port, la finalité étant d'arriver à un scan furtif, sans détection de cette activité sur la cible (ou, plus compliqué, sur le réseau de la cible et notamment les IDS).

Fyodor a débuté son exposé en présentant quelques fonctionnalités de NMAP, son outil scanner de port :

- Scan classique de type TCP Connect, facilement détectable dans les fichiers journaux du système
- Scan de type SYN, non détecté par un système classique, mais très visible sur un IDS
- Utilisation d'option « DECOY », permettant d'introduire d'autres adresses IP spoofées dans le scan. Il devient difficile de déterminer quelle est la machine ayant effectué le scan car elle est perdue au milieu d'une liste de 20 machines potentielles.

- Utilisation d'option de scan très lent… ainsi, le scan n'est pas détecté. (les IDS classiques ne peuvent pas conserver dans leurs tables d'états des scans durant plusieurs jours…).

D'autres modifications plus profondes peuvent être effectuées afin de rendre les scans NMAP encore plus difficilement détectables par les IDS. Ainsi, quand on regarde les règles de SNORT pour la détection de NMAP, certaines règles utilisent des caractéristiques réseau de NMAP : contenu du *payload, Window Size, data* dans les trames ICMP, *sequence number*… En recompilant NMAP après avoir modifié ces éléments, la détection devient vraiment plus délicate.

Une autre alternative consiste à essayer de crasher l'IDS. Dans le passé, certaines attaques ont été connues comme perturbatrices pour les IDS.

Fyodor a ensuite présenté deux outils permettant de jouer avec l'IDS « BlackIce » :

- Windentd : "*Simple demonstration program to spoof a Windows NMB name response -- primary purpose is to annoy/evade BlackICE Defender (and similar IDS products)"*

En effet, L'IDS Blackice, par défaut, émet une requête de demande de résolution de nom Windows lorsqu'une machine lève une alarme. Il est ainsi possible de spoofer les réponses en détournant le but initial de ce mécanisme. Lors de la démonstration par Fyodor, l'IDS effectuait sa résolution, et affichait des résultats complètement incohérents, de nature à perturber l'administrateur en charge de l'IDS.

- Icepick : Il s'agit d'un scanner d'IDS BalckIce. En tirant partie des caractéristiques actives de l'IDS BlackIce, Icepick est capable de les rechercher en scannant le réseau, et de déterminer leur version.

Fyodor a conclu son exposé en précisant que l'IDS choisie n'est qu'un exemple… BlackIce est parmi les très bons produits IDS, mais ceci montre qu'un outil de ce type doit être bien configuré (résolution Windows désactivé par exemple) sous peine de ne pas pouvoir tirer profit des possibilités du produit.

Le support de cette présentation est disponible à l'adresse :
http://www.insecure.org/presentations/CanSecWest01/index.html

3.3.5. The OpenBSD project

"Secure by default"

Theo de Raadt, deraadt@openbsd.org

Theo a présenté la philosophie et les actions du projet OpenBSD.

La philosophie du produit tient en une phrase : « **Secure by default** ». En effet, la plupart des produits sont livrés avec un nombre très important de fonctions et services activés par

défaut. (et tout le monde sait que les installations par défaut reste tel quel la plupart du temps…).

RedHat et Solaris sont des exemples de produit fournis avec un ensemble de démons importants fonctionnant par défaut. (cas de RedHat : linux mountd, linux rpc.statd...).

OpenBSD essaye de livrer des installations par défaut le plus sûr possible en désactivant tout ce qui n'est pas indispensable. Le challenge est de faire en sorte que l'activation de fonctions complémentaires soit simple.

Historique

Le projet a débuté en collaboration avec des personnes de SNI (Secure Networks, Inc) lors de la mise au point du scanner de sécurité Ballista. En collaboration avec SNI, les vulnérabilités reportées par Ballista étaient corrigées parallèlement sur OpenBSD.

L'étape suivante a fonctionné un peu de la même façon, en suivant les avis de sécurité de Bugtraq, et en corrigeant à la volée toutes le failles publiées.

Ensuite, le concept a été étendu à la correction systématique de toutes le failles semblables. Ainsi, suite aux publications de *bugtraq*, tous les démons ou programmes susceptibles d'avoir des problèmes étaient corrigés.

Fort de ces enseignements, le projet s'est alors orienté vers une approche beaucoup plus active, en corrigeant systématiquement toutes les erreurs et problèmes de programmation, afin de supprimer les failles de sécurité avant qu'elles n'apparaissent. La relecture et la modification de tout le code BSD est une tache longue et délicate, mais ce travail de fourmi porte ses fruits.

Mythe et réalité de l'OpenSource :

- Le code source est plus sûr car il est relu – FAUX, car peu de gens compétant relisent le code
- Les personnes utilisant du code source le corrigent et l'améliorent – FAUX, car les amateurs ont plutôt tendance à rajouter des failles de sécurité qu'à en enlever…
- L'amélioration du code source – La plupart des gens pensent encore : « Si ça marche, je n'y touche pas ! ».

Les travaux d'OpenBSD:

- Travaux sur la génération de nombres aléatoires pour les TCP ISN, IP ID…
- Travaux récents sur des appels critiques à la fonction select(2) fd_set

Actuellement, il reste encore des travaux à mener. L'exemple le plus récent concerne les appels à *glob(3)* dans la *glibc*.

OpenBSD souhaite avoir un rôle de « Guide » dans la communauté OpenSource. Les personnes intéressées, ainsi que les vendeurs peuvent regarder les solutions choisies par OpenBSD, et s'en inspirer afin de corriger les problèmes de sécurité de leurs produits.

Résultats d'OpenBSD :

Il y a eu quelques failles de sécurité dans BSD au cours des trois dernières années, mais aucune n'ont directement impacté le produit, car les éléments concernés étaient tous désactivés par défaut.

Relation entre OpenBSD et le reste du monde :

Avec Sendmail, il n'y a pas de problème, la collaboration est constructive.

Par contre, quelques soucis avec l'ISC et Bind. En effet, les versions 8 et 9 de BIND sont très obscures, et très mal sécurisées. En effet, l'ISC a pour politique de n'agir que sur les bugs de sécurité avérés, et non sur l'ensemble des problèmes présents (qui se révéleront très probablement être des failles de sécurité dans peu de temps).

Un projet de type « OpenDNS » pourrait éventuellement voir le jour…

Des informations complémentaires sont disponibles sur :

http://www.openbsd.org

3.3.6. Encryption - a 2 edged sword

Kurt Seifried - seifried@securityportal.com

Présentation classique de la cryptographie, de son utilisation, ainsi que de quelques problèmes associés.

http://www.securityportal.com

3.3.7. The road to Snort 2.0

3.3.7.1. L'avenir de SNORT (version GNU…)

Présentation de la roadmap de SNORT 2.0, par Martin Roesch, auteur de SNORT et Stuart Staniford, président de « Silicon Defense », une société fournissant un support commercial à SNORT.

Après avoir présenté globalement les fonctions de SNORT (combinaison d'un sniffer réseau, d'un analyseur de paquet et d'un NIDS), Marty a introduit les nouveautés de la version 1.8 (aussi appelé parfois 1.7.1…) :

- Nouveaux plugins
    - BO detection (Back Orifice)
    - TCP window size checking
    - passive OS identification

  - RPC normalization
  - telnet negociation code normalization
- Amélioration de plugins
  - Enhanced TTL detection
  - CVS output plugins
  - Enhanced database output plugin

D'autres améliorations ont pu être apportées à SNORT. Une fonction de marquage sera ainsi prochainement disponible. Ainsi, lors de l'activation d'une règle, une machine (basé sur son adresse IP) pourra être surveillée durant une période choisie (en seconde, nombre de paquet ou taille de donnés émises), et toutes les communications de ou vers cette machine pourront alors être journalisées.

D'autres fonctions sont actuellement à l'étude :

- Agrégation d'alertes
- Nouveau décodeur Unicode

Enfin, la nouvelle version disposera d'une base de règle complètement nettoyée et mise à jour.

Le future de SNORT

Après la sortie de la version 1.8, une nouvelle branche CVS démarrera le développement de SNORT 2.

Un travail important va être entrepris sur les plugins d'acquisition. En effet, la Libpcap, utilisée actuellement s'est avérée très utile, mais devient maintenant un point de blocage lors de la capture sur des réseaux haut-débits.

Le développement de nouvelles sondes permettrait d'augmenter sensiblement les capacités du produit (probablement au détriment de la portabilité, mais il n'y a pas d'autres alternatives…).

D'autres orientations sont à l'étude :

- Développement d'un routeur IDS (le routeur est à la bonne place pour analyser le trafic).
- Association d'un IDS et d'un firewall (reconfiguration plus facile de règles de sécurité en fonction de la nature du trafic…)
- Appliance IDS haute performance

Du coté des plugins, un effort sera fait pour l'ouverture du produit. D'autres protocoles pourraient être supportés dans le futur. Il serait même envisageable de modifier le moteur de détection sous la forme d'un nouveau plugin. En effet, le moteur couramment utilisé (à base de règles et de signatures) pourrait être étendu.

Du coté des règles, il pourrait être possible d'interfacer l'IDS avec une base de donnée contenant les règles. XML pourrait être supporté comme format de règles…

Un nouvel algorithme permettant d'améliorer la vitesse de l'analyse syntaxique (new pattern matcher) est aussi à l'étude.

Marty conclu alors en expliquant que les domaines de recherches ne manquent pas…

Fait d'actualité :

Commentaire obligatoire sur STICK…

Ce n'est pas la fin du monde… il existe plusieurs stratégies permettant de contrer ce type d'attaque, tels que l'agrégation d'alerte, ou des règles spécifiques de détection de ce type d'attaques…

3.3.7.2. L'avenir Commercial de SNORT

Martin Roesch et Stephen Northcutt viennent de fonder une société commerciale reposant sur l'utilisation de SNORT : Sourcefire Inc.

Cette société sera centrée sur le domaine des IDS, et fournira comme premier produit : « OpenSnort Sensor », une appliance SNORT de type 1U, fonctionnant sous OpenBSD…

Les détails complets de ce produit n'ont pas encore été révélés, mais on peut s'attendre à :

- Une interface Web d'administration
- Un éditeur de règle de SNORT
- Des viewers et éditeurs de fichiers journaux
- Un bon niveau de support…

Une lignée de produit suivra alors OpenSnort Sensor :

- Une console de management
- Des outils de support de réseau de SNORT (multi snort management)

Sourcefire s'appuiera fortement sur Silicon Defense pour le support de ces produits.

3.3.7.3. Spade and Spice (and Snort support).

Présentation de Stuart Staniford, Silicon Defense

Stuart a déjà effectué de nombreuses recherches sur les IDS, et il dirige actuellement un groupe de travail de l'IETF sur les IDS (CIDF).

SPADE (Statisitical Packet Anomaly Detection Engine) est un plugin de détection d'anomalies statistiques.

Il est très intéressant de consolider des résultats, tels que les portscans par exemple. Une représentation graphique selon les axes « adresse source », « adresse destination » et « port destination » permet par exemple de mettre en évidence des comportements anormaux.

Actuellement, le plugin SPADE ne s'intéresse qu'aux paquets de type SYN. Il attribue un score d'anomalie à des évènements [(X)=-log(P(X))], et reporte les événements dont le score dépasse un seuil prédéfinie.

Propriété des démarches statistiques :

- Difficile à configurer
- Présence de false positifs…
- Très dur à contourner
- Bonne complémentarité avec les techniques à base de signatures

Le module SPICE

SPICE sera un module complémentaire à SPADE, qui sera mis en ligne dans quelques mois. Il permettra de limiter les détections de false positives de SPADE, et proposera des représentations graphiques des résultats.

3.3.8. Snort Custom Rule Type

Présentation par Andrew R. Baker, de « FARM9 ».

Snort offre de nombreuses possibilités, et de nombreux paramétrage. Les « Snort custom rule types » en sont un exemple.

Par défaut, SNORT dispose de 5 actions prédéfinies, qui sont : Log, Pass, Alert, Activation & Dynamic.

La définition de nouveaux « ruletype » va permettre de commander des actions particulières sur évènements. Par exemple :

- Choix des plugins de sorties suivant le type d'alerte
- Choix du type d'action en fonction de l'alerte (et de l'ordre d'exécution des actions)
- Permet de mettre en place un type « nulle » permettant par exemple d'évaluer l'impact de nouvelles règles de filtrage sur un firewall, en journalisant des paquets activant les règles de ce type.

Par contre, ces nouvelles possibilités :

- Rajoute de la complexité au jeu de règles

- Peuvent entraîner des dégradations de performances

Ces fonctionnalités doivent donc être utilisées à bon escient.