



ekoparty 2007 – Argentina
Francisco Amato
evilgrade, *"You have pending upgrades..."*



Introduccion

Temario

- Client side exploitation
- Procedimiento de actualizacion
- Falla en la implementacion
- Escenarios de ataque
- Presentacion del framework evilgrade



Client side exploitation

Las politicas de seguridad generalmente de las compañías comienzan desde la periferia para luego atacar la seguridad interna.

Esta tecnica permite convertir los terminales de los usuarios en una puerta de acceso a la infraestructura interna de una compañía



Proceso general de actualizacion de una app

Como es?

- Actualizador es un proceso manual o automatico.
- Este proceso va a buscar a un server determinado un archivo con detalles por ejemplo `update.aplicacion.com/info.xml`
- Este archivo contiene informacion de los updates disponibles.
- Instala o pregunta si desea instalar los updates disponibles.



Cual es el problema?





Hay un problema?

Confianza

Muchas aplicaciones no verifican los updates.

Confían que el servidor que les entrega update es su servidor.



evilgrade

Descripcion

Es un framework modular que permite aprovecharse de los procesos de actualizacion de distintos aplicativos para inyectar updates falsos.

Esta desarrollado en Perl

Proyecto opensource



evilgrade

Como funciona?

Trabaja con modulos, cada modulo implementa la mecanica necesaria para emular updates falsos de una aplicación.

El framework necesita de la manipulacion del trafico dns que recibe la/s maquina/s cliente que se desea atacar.



Proceso normal

1. App1 inicia proceso de actualizacion
2. Consulta con el DNS server el host
update.app1.com
3. El DNS server devuelve 200.1.1.1
4. Aplicacion1 obtiene el archivo
<http://update.app1.com/lastupdate.xml>
5. App1 procesa archivo y detecta que hay un
update disponible
- 6 App1 baja y ejecuta el update
<http://update.app1.com/update.exe>



Ejemplo de ataque

1. App1 inicia proceso de actualizacion
2. Consulta con el DNS server el host update.app1.com
3. El atacante manipula el trafico DNS y devuelve otra direccion ip controlada por este.
4. Aplicacion1 obtiene el archivo manipulado por el atacante <http://update.app1.com/lastupdate.xml>
5. App1 procesa archivo y detecta que hay un update disponible
- 6 App1 baja y ejecuta el update <http://update.app1.com/backdoor.exe>



Cuales son los escenarios de ataque?

Posibilidades:

Escenario interno:

- *Acceso dns interno.*
- *Ataque ARP spoofing.*
- *Ataque DNS Caché Poisoning.*

Escenario externo:

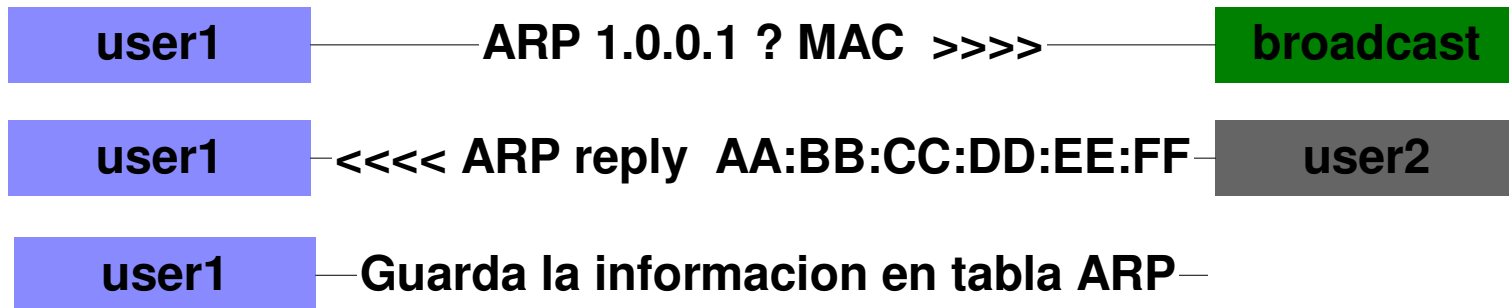
- *Acceso dns interno.*
- *Ataque DNS Caché Poisoning.*



ARP spoofing

Descripcion

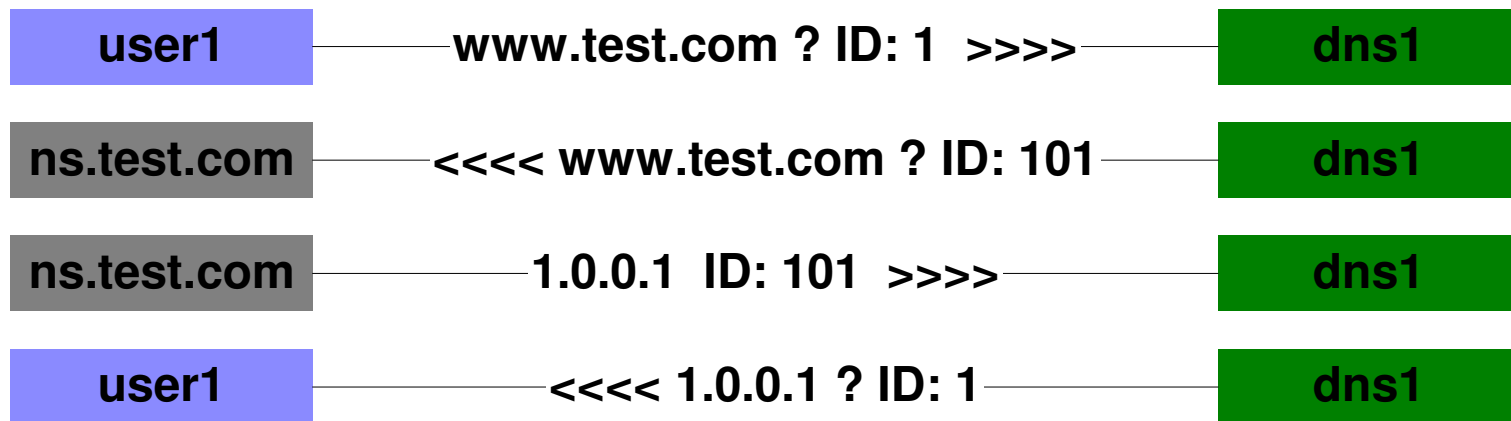
El protocolo ARP se utiliza para la resolucion de direcciones MAC utilizando una direccion ip.





DNS Cache poisoning

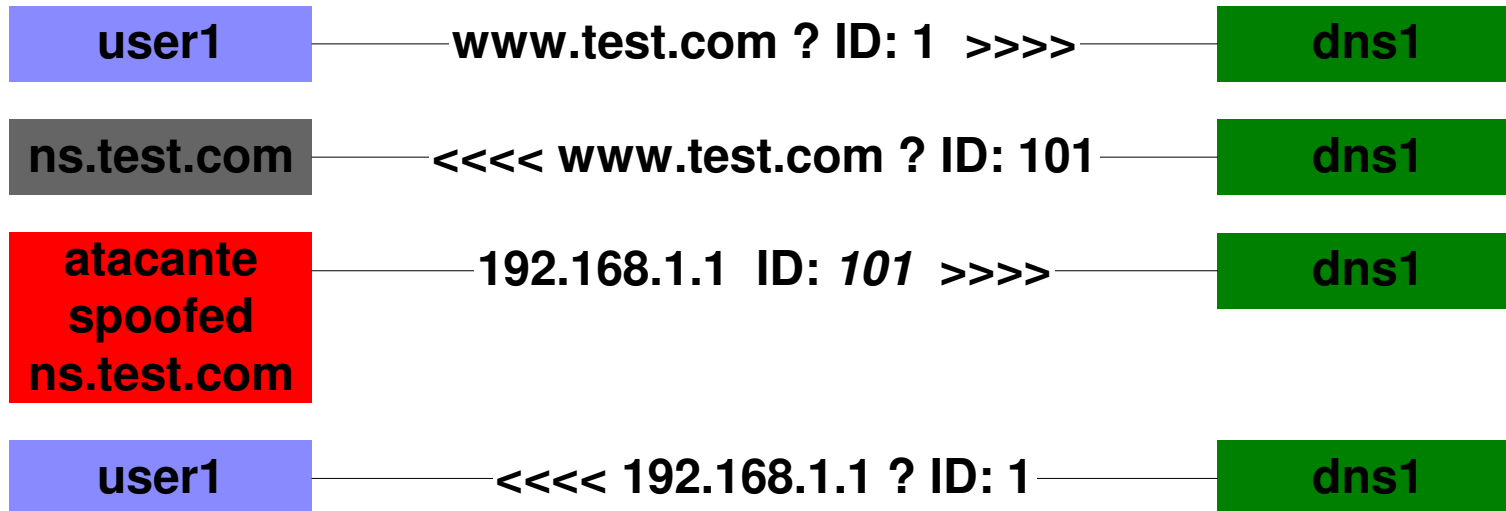
Descripcion





DNS Cache poisoning

Ataque





DNS Cache poisoning

Temas a tener en cuenta

A tener en cuenta:

- TTL
- No se encuentre cacheado.
- Respuesta legitima.

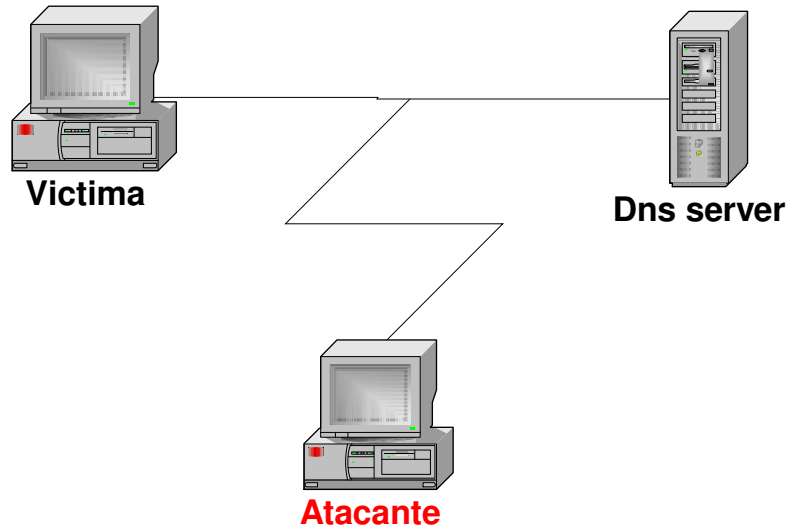
Datos necesarios:

- Source.
- ID 16 bits (65535 posibilidades).



Esquema interno

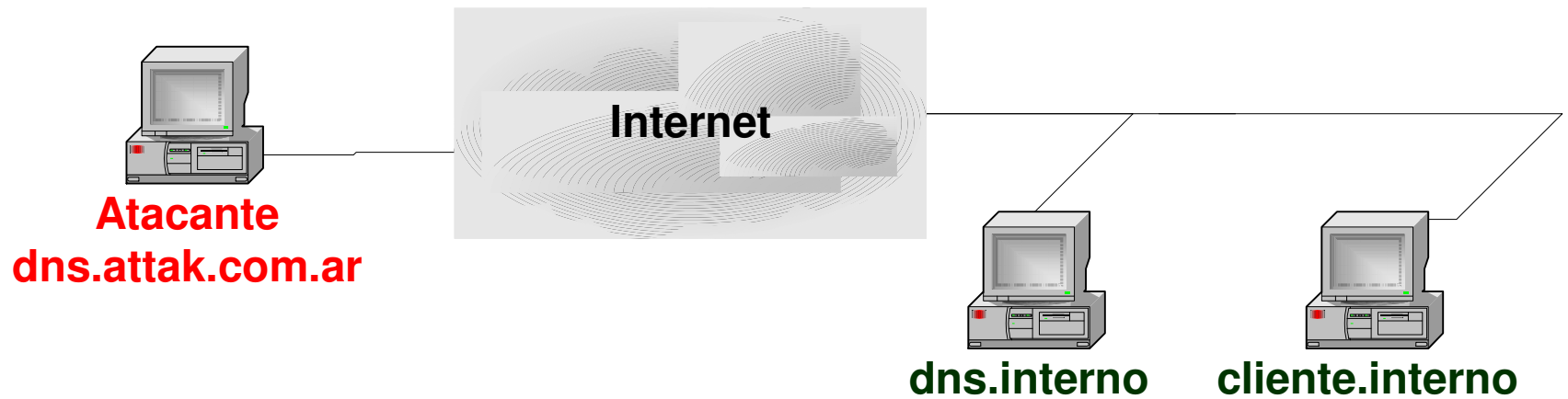
Descripcion





Esquema externo

Descripcion





Esto es nuevo?

No, esta tecnica ya es conocida.

La idea del framework es la centralizacion y explotacion de distintas falencias en la implementacion de sistemas de updates.



evilgrade

Que sistemas operativos soporta?

El framework es multiplataforma, solo debe contener el update falso correspondiente para la plataforma a explotar.



evilgrade

Para que me sirve?

Este vector de ataque permite el acceso al sistema por medio de la inyección de un update falso.



Consola:

Trabaja bajo consola, los comandos son simil IOS:

-show <object>: utilizado para mostrar informacion de distintos objetos.

-conf <object>: entrar en modo configuracion para un objeto.

-set <option> "value": configuracion de opciones.

-start: Inicia webserver.

-stop: Stop webserver.

-status: Estado de webserver.



Modulos:

```
package modules::sunjava;

use strict;
use Data::Dump qw(dump);

my $base=
(
    'name' => 'Sun Microsystems Java',
    'version' => '1.0',
    'author' => [ 'Francisco Amato <famato+[AT]+infobyte.com.ar>' ],
    'description' => qq(),
    'vh' => 'java.sun.com',
    'request' => [
        {
            'req' => '^/update/[.\d]+/map\-[.\d]+.xml', #regex friendly
            'type' => 'file', #file|string|agent|install
            'method' => '', #any
            'bin' => '',
            'string' => '',
            'parse' => '',
            'file' => './include/sunjava_map.xml'
        },
    ],
);
```



evilgrade

Request:

Es un colección de objetos.

Cada objeto corresponde a un http request posible dentro de la virtualhost asignado al modulo



Request:

Cada objeto contiene:

<req> - URL requerida (regex friendly).

<type> : [file | string | agent | install]

<method> : [GET|POST|TEST|""]

<bin> : [1|""] Si es un file binario.

<string> : Texto respuesta del request

<parse> : [1|""] Si desea ser parseado el string/file

<file> : path de archivo de respuesta al request



Modulos:

```
'options' => { 'agent' => { 'val' => './agent/reverseappsign.exe',
                        'desc' => 'Agent to inject'},
              'arg' => { 'val' => '',
                        'desc' => 'Arg passed to Agent'},
              'enable' => { 'val' => 1,
                            'desc' => 'Status'},
              'title' => { 'val' => 'Critical update',
                            'desc' => 'Title name display in the update'},
              'description' => { 'val' => 'This critical update fix internal vulnerability',
                                'desc' => 'Description display in the update'},
              'atitle' => { 'val' => 'Critical update',
                            'desc' => 'Title name display in the systray item pop'},
              'adescription' => { 'val' => 'This critical update fix internal vulnerability',
                                  'desc' => 'Description display in the systray item pop'},
              'website' => { 'val' => 'http://java.com/moreinfoLink',
                             'desc' => 'Website display in the update'}
            }
```



evilgrade

Agente:

Se le llama al update falso que va a ser inyectado en el equipo victima.



evilgrade

Modulos implementados:

- Java plugin
- Adobe Acrobat
- DAP (Download Accelerator).



Alfin!:

No quiero que se duerman...





evilgrade

Diseño mas seguros para procedimiento updates

- Servidor update bajo https, control certificado valido sobre el dominio emitido.
- Firma digitales, verificando el update con la clave publica distribuido localmente



Para la proxima!

Quando realicen un update



infobyte



No se dejen llevar por las apariencias





Preguntas?

???



Referencias

Title

- <http://www.secureworks.com/research/articles/dns-cache-poisoning/#update>
- <http://www.trusteer.com/docs/bind9dns.html>
- <http://www.trusteer.com/docs/bind8dns.html>
- http://en.wikipedia.org/wiki/ARP_spoofing
- <http://www.trusteer.com/docs/microsoftdns.html>



Gracias!

Contacto

Francisco Amato – famato@infobyte.com.ar

infobyte