CS 155
Homework 2

Rafael Moreno Ferrer
SUID#: 05490330

Problem 1. *(HTTPS vs. Signatures)*

Suppose a software company widgets.com sells a product P and wants to distribute a software update D. The company wants to ensure that its clients only install software updates published by the company. They decide to use the following approach: The company places D on its web server and designs the software P to periodically check this server for updates over HTTPS.

a) The company decides to buy a public key certificate for its web server from a reputable CA. Explain what checks P should apply to the server's certificate to defeat a network attacker. Is your design vulnerable to ssl strip?

b) How would you design the program P and the company's web server so that the update is secure against a network attacker, but there is no need to buy a certificate from a public CA? Your design should use an HTTPS web server as before.

c) Later on engineers at the company proposed the following very different design: Sign D using a widgets.com private key to obtain a signature s and then distribute (s,D) in the clear to all customers. The corresponding public key is embedded in the program P.

> *i) Later on engineers at the company proposed the following very different design: Sign D using a widgets.com private key to obtain a signature s and then distribute (s,D) in the clear to all customers. The corresponding public key is embedded in the program P.*
>
> *ii) How much computing time does widgets.com web server spend doing crypto calculations in each of the designs? Which is better?*

Problem 2. *(Stealth Port Scanning)*

Recall that the IP packet header contains a 16-bit identification field that is used for assembling packet fragments. IP mandates that the identification field be unique for each packet for a given (SourceIP,DestIP) pair. A common method for implementing the identification field is to maintain a single counter that is incremented by one for every packet sent. The current value of the counter is embedded in each outgoing packet. Since this counter is used for all connections to the host we say that the host implements a global identification field.

a) Suppose a host P (whom we'll call the Patsy for reasons that will become clear later) implements a global identification field. Suppose further that P responds to ICMP ping requests. You control some other host A. How can you test if P sent a packet to anyone (other than A) within a certain one minute window? You are allowed to send your own packets to P.

b) Your goal now is to test whether a victim host V is running a server that accepts connections to port n (that is, test if V is listening on port n). You wish to hide the identity of your machine A and therefore A cannot directly send a packet to V, unless that packet contains a spoofed source IP address. Explain how to use the patsy host P to test if V accepts connections to port n.
Hint: Recall the following facts about TCP:
– A host that receives a SYN packet to an open port n sends back a SYN/ACK response to the source IP.
– A host that receives a SYN packet to a closed port n sends back a RST packet to the source IP. A host that receives a SYN/ACK packet that it is not expecting sends back a RST packet to the source IP.
– A host that receives a RST packet sends back no response.

Problem 3. *(WPAD)*

WPAD is a protocol used by IE to automatically configure the browser's HTTP and HTTPS proxy settings. Before fetching its first page, IE will use DNS to locate a WPAD file, and if one is found, will use its contents to configure IE's proxy settings. If the network name for a computer is pc.cs.stanford.edu the WPAD protocol iteratively looks for wpad files at the following locations:

```
http://wpad.cs.stanford.edu/wpad.dat
http://wpad.stanford.edu/wpad.dat
http://wpad.edu/wpad.dat     (prior to 2005)
```

a) Explain what capabilities were inadvertently given to the owner of the domain wpad.edu as a result of this protocol. Explain how personal information can be exposed as a result of this issue.

b) Are pages served over SSL protected from the problem you described? If so, explain why; if not, explain why not.

Problem 4. *(CSRF Defenses)*

a) In class we discussed Cross Site Request Forgery (CSRF) attacks against web sites that rely solely on cookies for session management. Briefly explain a CSRF attack on such a site

b) A common CSRF defense places a token in the DOM of every page (e.g. as a hidden form element) in addition to the cookie. An HTTP request is accepted by the server only if it contains both a valid HTTP cookie header and a valid token in the POST parameters. Why does this prevent the attack from part (a)?

c) One approach to choosing a CSRF token is to choose one at random. Suppose a web site chooses the token as a fresh random string for every HTTP response. The server checks that this random string is present in the next HTTP request for that session. Does this prevent CSRF attacks? If so, explain why. If not, describe an attack.

d) Another approach is to choose the token as a fixed random string chosen by the server. That is, the same random string is used as the CSRF token in all HTTP responses from the server over a given time period. You may assume that the time period is chosen carefully. Does this prevent CSRF attacks? If so, explain why. If not, describe an attack.

e) Why is the Same-Origin Policy important for the cookie-plus-token defense?

Problem 5. *(Firewalls and Fragmentation)*

The IP protocol supports fragmentation, allowing a packet to be broken into smaller fragments as needed and re-assembled when it reaches the destination. When a packet is fragmented it is assigned a 16-bit packet ID and then each fragment is identified by its offset within the original packet. The fragments travel to the destination as separate packets. At the destination they are grouped by their packet ID and assembled into a complete packet using the packet offset of each fragment. Every fragment contains a one bit field called "more fragments" which is set to true if this is an intermediate fragment and set to false if this is the last fragment in the packet. We discussed the way small offsets can allow the data in one fragment to overlap information in the preceding fragment. However, overlapping fragments should not occur in normal network traffic.

In class we mentioned that when fragments with overlapping segments are re-assembled at the destination, the results can vary from OS to OS. Give an example where this can cause a problem for a network-based packet filtering engine that blocks packets containing certain keywords. How should a filtering engine handle overlapping fragments to ensure that its filtering policy is not violated?

Problem 6. *(DNSSEC)*

DNSSEC (DNS Security Extensions) is designed to prevent network attacks such as DNS record spoofing and cache poisoning. Generally, the DNSSEC server for example.com will posses the IP address of www.example.com. When queried about this record that it possesses, the DNSSEC server will return its answer with an associated signature. If the DNSEC server is queried about a host that does not exist, such as doesnotexist.example.com, the server uses NSEC or NSEC3 to show that the DNS server does not have an answer to the query.

Suppose a user R (a resolver, in DNS terminology) queries a DNSSEC server S, but all of the network traffic between R and S is visible to a network attacker N. The attacker N may read requests from R to S and may send packets to R that appear to originate from S.

a) Why is authenticated denial of existence necessary? To answer this question, assume that S sends the same unsigned DOES-NOT-EXIST response to any query for which it has no matching record. Describe a possible attack.

b) Assume now that S cryptographically signs its DOES-NOT-EXIST response, but the response does not say what query it is a response to. How is an attack still possible?

c) A DNSSEC server may send a signed NSEC response to a query that does not have a matching record (such as doesnotexist.example.com). An NSEC response contains two names, corresponding to the existent record on the server that immediately precedes the query (in lexicographic order), and the existent record that immediately follows the query. For example, if a DNSSEC server has records for a.example.com, b.example.com, and c.example.com, the NSEC response to a query for (non-existent) abc.example.com contains a.example.com and b.example.com because these come just before and just after the requested name. To be complete, NSEC records also wrap-around, so a query for a non-existent name after the last existent name will receive an NSEC containing the last and first existent names.

How should the resolver use the information contained in NSEC records to prevent the attacks you described in previous parts of this problem?

d) NSEC leaks information that may be useful to attackers on the Internet. Describe how an attacker can use NSEC to enumerate all of the hosts sharing a common domain-name suffix. How is this information useful for attackers?

e) NSEC3 is designed to prevent DNS responses from revealing unnecessary information. NSEC3 uses the lexicographic order of hashed records, instead of their unhashed order. In response to a query without a matching record, NSEC3 will return the hashed names that are just before and just after the hash of the query. For example, on a server containing a.example.com, b.example.com, and c.example.com, if a hashes to 30, b to 10, c to 20, and abc to 15, the NSEC3 response to a query for abc.example.com would contain 10.example.com and 20.example.com. Hashed names are also assumed to wrap around, in the same way as unhashed names in NSEC.

Explain how a resolver should verify the validity of a response under NSEC3?