Appendix C: Session Tracking Best Practices

Session tracking consists of two components:

- the session ID itself (e.g. the properties of unique identifier),
- and the transport mechanism (e.g. how the session ID gets transmitted between the browser and server).

This appendix contains two sets of best practices, one for the transport mechanism and one for the session ID.

Transport Mechanism

Cookies are currently one of the best ways to transport a session ID between browsers and servers/applications. Cookies used for session IDs should have the following characteristics:

- ❑ marked SECURE (i.e. sent with encryption)
- ❑ non-persistent (i.e. not stored on user's hard drive)
- ❑ reasonably limited in their Path and Domain
- ❑ keep session ID information out of URLs (this is not an issue for cookies, but does apply if URL rewriting is used to transport session ID instead of cookies)

Session ID

Strong session IDs should have the following properties:

❑ Not Related to User Information

Make sure the session ID is not related to user information. That's a fundamental aspect of a session ID, it is a short-term secret, thus any relationship to user information would violate that fundamental property. Besides, if a user suspects their social security number is embedded in a cookie you send them, heads will roll when they post their privacy concerns to the Internet.

❑ Randomness

The session ID generated by the web server / application should not exhibit any predictable patterns.

❑ Size

Even if a session is random, if it is small an attacker could brute force attack it. A 32-byte alphanumeric string should be sufficient.

- ❑ Perishable

Eventually expires, and cannot be reused/replayed (short-term vs. long-term secrets).

- ❑ Secure Transport

Sent over a secure path - to prevent eavesdropping.

- ❑ Tamper Prevention

Cryptographic properties, like checksum digits, to prevent tampering. Why would someone tamper with there own cookie? They may want to impersonate another user.

Appendix D: Application Session ID Analysis

> *Below is the text from a report written for a customer of Maven Security Consulting Inc. Certain names have been changed to protect the confidentially of our customer's identity.*

The session ID used by Reservation Portal can be predicted and used to view (and possibly) modify the itineraries of other users. Here is a typical session tracking cookie from Reservation Portal:

SKIstats=MessageID=&UniqueID=20030227A
A587885A12561A40&SiteID=1

The session ID within the SKIstats cookie is called UniqueID.

Here is a sampling of session IDs collected from Reservation_Portal.ski.dom:

```
20030227A161015A756635A12785A42
20030227A161017A617809A12252A41
20030227A161028A612536A12255A41
20030227A161031A719721A12790A40
20030227A161034A596716A12256A41
20030227A161037A839293A12210A39
20030227A161038A730268A12788A42
20030227A161039A735541A12789A42
20030227A161042A844567A12211A39
20030227A161044A828747A12212A39
20030227A161046A709174A12792A40
20030227A161046A719721A12790A42
20030227A161049A714448A12793A40
20030227A161052A601989A12257A41
20030227A161054A709174A12792A42
20030227A161057A698627A12794A40
20030227A161058A714448A12793A42
20030227A161100A698627A12794A42
20030227A161104A703901A12795A40
20030227A161105A703901A12795A42
20030227A161110A688080A12796A42
20030227A161113A688080A12796A40
20030227A161113A693354A12797A42
20030227A161120A677534A12798A40
20030227A161122A823473A12215A39
20030227A161122A464839A12499A43
20030227A161128A682807A12799A40
20030227A161130A807653A12216A39
20030227A161131A456008A12800A42
20030227A161136A456008A12800A40
```

Figure 0-1 Reservation Session IDs

At first appearance these session IDs seem robust. However, closer inspection reveals that the numbers are highly predicable.

Each session ID is composed of five groupings, each grouping is separated by the letter 'A'.

Group 1 (G1): The date the session ID was created (Year Month Day)

Group 2 (G2): The time the session ID was created (Hours Minutes Seconds)

Group 3 (G3): A six digit number

Group 4 (G4): A five digit number

Group 5 (G5): A two digit number

When we sort the session IDs by their G4 values, we see that G4 values are occurring in approx. four series (i.e. clusters or sub groupings).

Below we see one particular series of G4 values. The data has been separated into the groupings defined above to help reveal the pattern.

Date	Time	G3	G4	G5
20030227	161015	756635	12785	42
20030227	161038	730268	12788	42
20030227	161039	735541	12789	42
20030227	161031	719721	**12790**	40
20030227	161046	719721	**12790**	42
20030227	161046	709174	12792	40
20030227	161054	709174	12792	42
20030227	161049	714448	12793	40
20030227	161058	714448	12793	42
20030227	161057	698627	12794	40
20030227	161100	698627	12794	42
20030227	161104	703901	12795	40
20030227	161105	703901	12795	42
20030227	161113	688080	12796	40
20030227	161110	688080	12796	42
20030227	161113	693354	12797	42
20030227	161120	677534	12798	40
20030227	161128	682807	12799	40
20030227	161136	456008	12800	40
20030227	161131	456008	12800	42

Figure 0-2 Reservation Session ID Pattern for Give G4 Series

One observes that G4 values are sequential (X, X+1, X+2, etc). Furthermore, there is typically one value for G5 associated with any G4 series. In this example, G5 is normally equal to 42. However, if the G4

value repeats, then a second G5 value is used. For example, G4 equals 12790 in two session IDs above. G5 is 40 for one, and 42 for the other.

In addition, analysis shows that **G4 is not dependent on time**. Maven Security was able to collect sequential G4 values (X and then X+1) regardless of the time delay between requests.

Therefore, if a particular G4 value is missing, it was assigned to another user's session. **This key fact** allows an attacker to begin analyzing session IDs in order to hijack other user sessions. The steps required are described below:

> *The following data was collected rapidly as an unauthenticated anonymous user repeatedly requesting the first page of the reservation process. Meanwhile, Maven Security used a test account to simulate a victim user. This user completed a reservation online. The analysis (documented below) was performed to determine if a 3rd party could guess the session ID of the legitimate user. In order to avoid affecting real customers, a test account (i.e. simulated victim) was used by Maven Security.*

1) Collect session IDs through repeated requests to http://Reservation Portal.ski.dom

2) Split the session IDs into the five groupings or columns (delimited by the letter 'A'). Then sort them by G4 (i.e. the 4th column of numbers). Find a missing G4 value. **Because G4 values are sequential, this is trivial.** The attacker now knows the G4 value for another user. In this example, we are using **12787** since it is missing from the values collected. See figure below. We also are using 12787 because this value belongs to the simulated victim. Other missing values of G4 belong to real customers, and will not be tampered with out of fear of disrupting real customers.

3) Guess the G5 value associated with the missing G4. G5 can normally be any one of five possible values throughout all the data, but only two are used within a clustering of G4. And even then, one value of G5 is typically used unless the value for G4 repeated. Therefore, one value of G5 is considered dominate and would be the first choice to try. In this example we are using **42** because that is the most common value seen for the G4 values within the clustering (below).

4) Ignore all other session IDs except those within the G4 cluster of interest (where the missing value would have been). Also, ignore session IDs that don't have the G5 value picked in the previous step. The data we are left with is shown below. The date column (G1) has been removed for brevity (besides, it is static and known being equal to the current date – therefore, G1 does not need to be included at this point for analysis).

ITEM	G2 (Time)	G3	G4	G5	Calculated G3 Deltas
B			12786 <- Missing G4		
C	[Need time]	[Calc. G3]	**12787 <- Missing G4**		
D	**161038**	730268	12788	42	Calculate G3 Delta (-15820 or -15821)
E	161039	735541	12789	42	5273
F	161046	719721	12790	42	-15820
G			12791 <- Missing G4		
H	161054	709174	12792	42	
I	161058	714448	12793	42	5274
J	161100	698627	12794	42	-15821
K	161105	703901	12795	42	5274
L	161110	688080	12796	42	-15821
M	161113	693354	12797	42	5274
N	161131	456008	12800	42	-237346

Figure 0-3 Cracking Reservation Sessions

5) Calculate the difference (delta) between sequential G3 values. A pattern becomes clear. The delta alternates between two values (-15821 and 5274). Actually, each delta value seems to occasionally fluctuate by one. Therefore, we are able to calculate the two possible values for the missing G3 (associated with the missing G4) by applying the deltas. **G3 is either 746088 or 746089**.

6) Examine the time gap in the collected data. The missing G4 was issued somewhere within that window of time. In this example, the time (hour min sec) goes from **161015** (the last G4 we saw before a G4 value was seen to be missing) and **161038**. That leaves 22 possibilities, one for each second between those time stamps. The victim user who was given the G4 value of 12787 (the one targeted by this example) was issued that session ID somewhere within this time window of 22 seconds. The faster we collect session IDs in step 1, the smaller this time window will be. The attacker now knows that G2 (time) is between **161015** and **161038**

7) Finally, the session ID must be associated with a SiteID (another cookie value with SKIstats that represents the resort location, such as Suicide Slope versus Widow Maker). A session ID (UniqueID) must be paired with the correct SiteID in order for the attacker to be able to hijack the session. This leaves the attacker with four possibilities for the SiteID cookie.

All of the required combinations documented above are summarized below:

- G1 (Date) is known.
- G2 (Time) has 22 possibilities (between **161015** and **161038**). This can be reduced if session IDs are collected quickly in step 1.
- G3 is calculated to be **746088** or **746089** (2 possibilities due to slight fluctuation in G3 deltas). However, the fluctuation does not seem random, so one choice would be more favored.
- G4 is known due to the gap in our collected data. There were several missing G4 values, each representing the session ID for other users. We targeted **12786.**
- G5 is predictable by the series (or clustering) from which the targeted G4 is located. Technically there are two possibilities for G5 given any G4, although one could argue that one value is dominant. The value for G5 is known to be **42** in this case.
- SiteID is one of four possibilities (between 1 and 4 inclusive).

Therefore, the total number of requests needed to hijack another user's Reservation Portal session is 352 (= 22 * 2 * 2 * 4). This can be reduced to 176 if you think step 3 has one dominate choice; or even as low as 88 if you accept that the random delta fluctuation in step 5 is predictable. And of course, the faster the attacker collects cookies in step one, the smaller the time gap in step 6 – and that will quickly lower the total number of guesses required.

> *The above steps were used to calculate a test victim user's session ID, and view their completed itinerary.*
>
> *The session ID of an actual Reservation Portal user was not used. A test account was used to avoid affecting real customers.*

The impact of this finding for the Reservation Portal site is that **an attacker could view** (and possibly modify) the reservation **itineraries of other users**.

The biggest threat in this case is the possibility of Broken Leg Ski Resort adding more functionality to the Reservation Portal site, or using the same session ID technique for other applications.