Modeling Wi-Fi Protected Setup Brute-Force Mitigations Using Markov Chains

Lloyd W. Jones
Radford University
Radford, VA
ljones9@radford.edu

I. Introduction

In an attempt to cater to non-technical users, most consumer and small business wireless routers include a feature known as Wi-Fi Protected Setup, or WPS. Developed by the Wi-Fi Alliance, this feature enables users to connect a wireless device to an access point by pushing a button or entering a PIN number, as opposed to remembering a typically long or difficult to memorize passphrase. The weak security of the WPS protocol is already well-known and has been documented by multiple individuals. Additionally, tools have been developed to automate the exploitation of this technology. The goal of this paper is to examine the timeout and lockout mechanisms that some routers employ and determine their overall effectiveness in delaying brute-force attacks on the WPS protocol. Using Markov chains, we can model the probability of a successful attack at different stages of the WPS authentication process. We can then make a decision as to which method produces the best results. This topic is interesting because of the fact that WPS is so widely deployed despite its inherent security weaknesses. Also, I have not found any studies concerning brute-force mitigations mechanisms in wireless routers.

II. Problem Statement

Each stage of the WPS authentication process presents a fixed probability of a host guessing the information needed to proceed to the next stage. Modeling these probabilities with Markov chains, we can visually understand how this works. Afterwards, we can choose different delay values and calculate the probability of a successful brute-force attack given a certain amount of time. We will analyze common values for these variables based on real-world testing and analysis of consumer wireless routers. For example, we can show the probability of a successful attack taking place within five hours given an access point lockout of 60 seconds, a PIN verification time of 4 seconds per PIN, and a lockout of 60 seconds per 3 failed PIN attempts. In our analysis, we will use the following variables:

$0 \leq d \leq 5$	Access point-imposed delay between PIN attempts (seconds)
$0 \leq s \leq 120$ $3 <= p <=25$	Lockout (s) in seconds per amount of consecutive incorrect PINs (p)
$t \geq 0$ in	Time limit for successful attempt (minutes)
$0 \leq x \leq 11{,}000$	Number of total PIN attempts possible given t,v,d,s, and p
$0 \leq v \leq 5$	Access point PIN validation time in seconds
$0 \leq P_0 \leq 1$	Probability of client being in unauthenticated state
$0 \leq P_1 \leq 1$	Probability of brute-forcing first half of PIN
$0 \leq P_2 \leq 1$	Probability of brute-forcing second half of PIN
$0 \leq P \leq 1$	Overall probability of successful brute-force given d, $L_{(s/p)}$, t, v

Figure 1. Variables and Assumptions

III. Literature Survey

The WPS standard was originally published in December of 2006 [1] and has been implemented in wireless routers sold by a wide array of manufacturers. Securing devices such as these can seem intimidating to end users, so the WPS protocol was established to meet this need. Instead of memorizing a WEP or WPA passphrase, a user simply needs to push a button on the access point and the device in order to pair them. Alternatively, a PIN code can be generated by the access point and then entered into the device to accomplish the same goal. In December of 2011, Stefan Viehboch came across a vulnerability in the implementation of WPS which allowed the brute-forcing of the WPS PIN, a required feature of this specification. WPS is usually enabled by default on routers that support it. Even worse, some devices do not offer the options to disable it. Others allow the user to disable WPS but don't actually turn it off at all. Open-source tools are now freely available to easily exploit this vulnerability.

The aforementioned attack is made possible by a weakness in PIN validation between the enrollee, or client device requesting access, and the registrar, or the device providing wireless settings to the enrollee. The format of the PIN is shown below [3]:

1	2	3	4	5	6	7	Checksum
First half of PIN				Second half of PIN			

Figure 2. Wi-Fi Protected Setup PIN Format

Ideally, this numeric implementation should allow for 10^7 PINs, but instead the PIN is split into two halves and each one is verified separately. This behavior is shown in the M4-M7 messages included in the WPS specification document [2].

<table>
<tr><td>M4</td><td>Enrollee → Registrar</td><td>N1||R-Hash1||R-Hash2||$E_{\text{KeyWrapKey}}$(R-S1)|| Authenticator</td></tr>
<tr><td>M5</td><td>Enrollee ← Registrar</td><td>N2||$E_{\text{KeyWrapKey}}$(E-S1)||Authenticator</td></tr>
<tr><td>M6</td><td>Enrollee→ Registrar</td><td>N1||$E_{\text{KeyWrapKey}}$(R-S2)||Authenticator</td></tr>
<tr><td>M7</td><td>Enrollee ← Registrar</td><td>N2||$E_{\text{KeyWrapKey}}$(E-S2||ConfigData)||Authenticator</td></tr>
</table>

Figure 3. Authentication between Enrollee and Registrar

This detail drastically reduces the amount of possible PINs from 10^7 (10,000,000) to 10^4 + $10^{3\,(}$11,000). If the correct PIN is found, the registrar will provide the enrollee with the WPA/WPA2 encryption key. Armed with this information, an attacker is able to authenticate with the access point and access the network.

As stated above, some of these devices include a timing delay or lockout after a PIN has been incorrectly entered a certain number of times. This functionality is not documented or required by the WPS specification. Additionally, router manufacturers do not include information about these mechanisms online or elsewhere. Devices can have no delay, a static delay, an incremental delay, or total lockout. Brute-force attempts may also be influenced by other factors as well, including signal strength, interference, access point load, and client wireless interface limitations. We will examine multiple scenarios and attempt to model reasonable values based on real-world data while comparing and contrasting the various methods.

IV. Problem Solving Approach

The approach used to solve this problem will be to first model the probabilities of each PIN half being successfully guessed in a scenario with no delay or lockout mechanism in place and with no external factors affecting the PIN exchange process. Afterwards, we will examine and document commonly observed lockout mechanisms in consumer wireless routers. We will then compute, model, and graph the success percentages of brute-force attacks against WPS in each system given various timeframes. We can then compare and contrast the various approaches and determine their overall effectiveness. It is important to note that some tools used in the WPS brute-forcing process do not perform the process incrementally, but instead start with certain manufacturer specific or commonly seen hard-coded PINs instead. In our analysis, we make no such assumption.

In order to model these probabilities, we will use the following equations to compute the probability of a successful brute-force attempt:

$$x = \frac{t}{(v+d)(\frac{s}{p})} \tag{1}$$

$$\mathrm{P} = \sum_{n=0}^{x} \frac{1}{11000-n} \tag{2}$$

$$P_1 = \sum_{n=0}^{x} \frac{1}{10000-n} \tag{3}$$

$$P_2 = \sum_{n=0}^{x} \frac{1}{1000-n} \tag{4}$$

The first equation (1) calculates the number of PINs that can be brute-forced given a time constraint in seconds. This is done by dividing the time limit in seconds by the sum of the PIN verification/validation time of the access point plus the per-PIN delay of the access point multiplied by the number of seconds of delay divided by the amount of incorrect PINs needed for the delay. The second equation (2) then uses this result to compute the summation of each probability, giving us the overall chance of success to brute-force the entire PIN. Similarly, equation three (3) does this but only for the first half of the PIN. Lastly, equation four (4) is the same but for the second half of the PIN.

$$\mathrm{P} = \begin{bmatrix} P_0 \to P_0 & P_0 \to P_1 & P_0 \to P_2 \\ P_1 \to P_0 & P_1 \to P_1 & P_1 \to P_2 \\ P_2 \to P_0 & P_2 \to P_1 & P_2 \to P_2 \end{bmatrix}$$

Figure 4. Markov Chain Representation

The above matrix notation is used to depict Markov chains relating to this problem. The three states used are unauthenticated/associate (P_0), first half of PIN correct (P_1), and second half of PIN correct or successfully authenticated (P_2).

V. Timeline

- April 11 – Submit proposal.
- April 11-13 – Research common router lockout and delay periods. Model Markov chains.
- April 13-17 – Perform experiments and computations. Analyze and graph results.
- April 17-20 – Write paper detailing work in IEEE transaction format.
- April 20-25 – Create presentation slides of work done.
- April 26 – Submit final paper and presentation.
- April 30- May 2 – Give presentation to class

VI. RESULTS

An initial baseline was first created to better understand subsequent results. This baseline was generated from a “best case scenario” understanding of the problem. The results of running reaver, a popular WPS brute-forcing tool, were analyzed to obtain real-world information about popular consumer wireless routers. From this, we found a minimum PIN processing/verification time of 3 seconds with no per-PIN delay and no lockout (based on the Linksys E1000). These baseline results are modeled below assuming a maximum brute-force time of 5 hours:

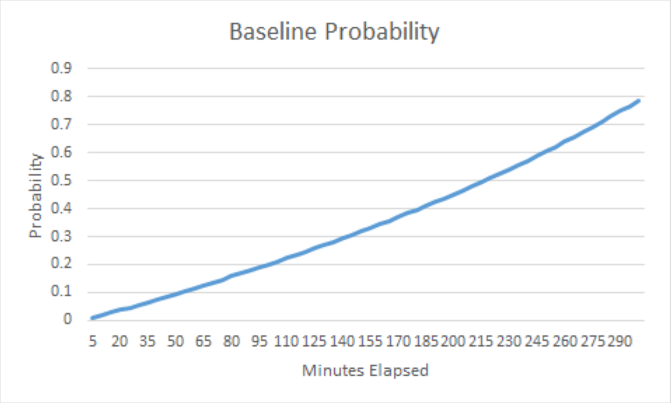

Figure 5. Baseline Probabilities

$$P = \begin{bmatrix} .09 & .91 & 0 \\ 0 & .01 & .99 \\ 0 & 0 & 0 \end{bmatrix}$$

As we can see, there was a very high chance (about 79%) that the PIN could be recovered in five hours using this baseline scenario. There was roughly a 25% chance of success at 2 hours, a 50% chance of success at 3.5 hours, and a 75% chance at 4.8 hours. The Markov chain representation showed that there was a 91% chance of successfully recovering the first half of the PIN within 5 hours and a 99% chance of recovering the second half of the PIN as well. Armed with this information, we can now see how mitigation mechanisms change the results.

We then moved on to per PIN delay mitigations. It has been observed that one of the most common per-PIN delay configurations is 5 seconds per PIN. Using the same PIN processing time as above (2 seconds), we modeled this setup using the following graph and Markov chain representation:

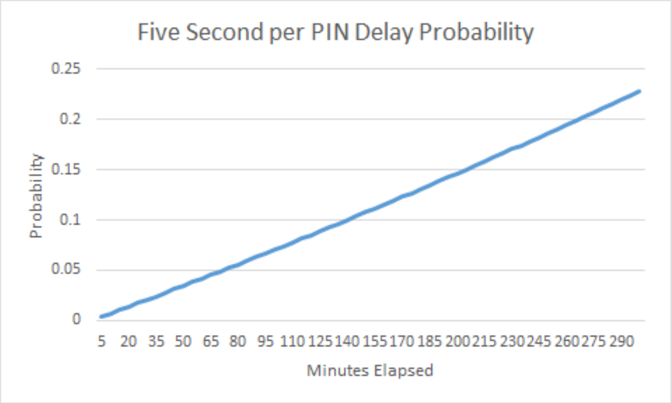

Figure 6. Five Second per PIN Delay Probabilities

$$P = \begin{bmatrix} .75 & .25 & 0 \\ 0 & .44 & .56 \\ 0 & 0 & 0 \end{bmatrix}$$

Compared to the baseline, it was easy to see the impact of a per PIN delay. Our overall success rate within 5 hours dropped to 22%, with a 10% chance of success within 2.3 hours and a 20% chance of success within 4.5 hours. The first and second PIN half success rates also dropped significantly to 25% and 56% respectively. The effectiveness of such a method seems obvious given these numbers, but we can still conclude that this mitigation won’t phase an attacker who has plenty of time on their hands.

For the next test, we modeled a longer per PIN delay by setting d = 10 to see how it affected the probabilities:

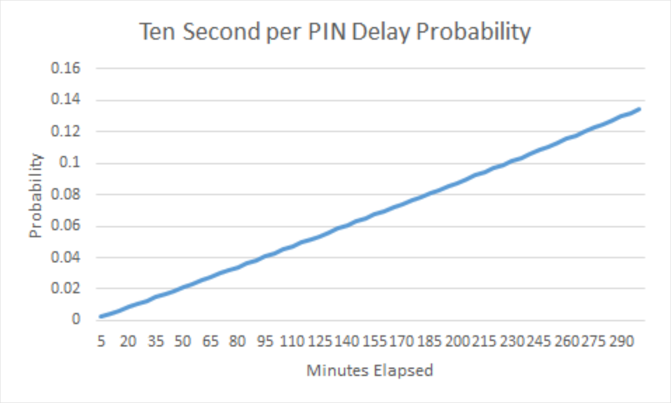

Figure 7. Ten Second per PIN Delay Probabilities

$$P = \begin{bmatrix} .85 & .15 & 0 \\ 0 & .79 & .21 \\ 0 & 0 & 0 \end{bmatrix}$$

Yet again, we experienced as significant drop in probabilities, as expected. We found a 13% overall success rate with a 15% success rate for the first half of the PIN and a 21% success rate for the second half of the PIN.

Next, we compared the PIN lockout method to the previous results. After researching documented access point lockouts, we were able to model average values for both *s*, the number of seconds, and *p*, the number of failed PINs needed to activate the lockout period. Below are the results of probability modeling using *p=10* and *s=30*. For the sake of continuity, we still used a PIN processing time of 2 seconds:

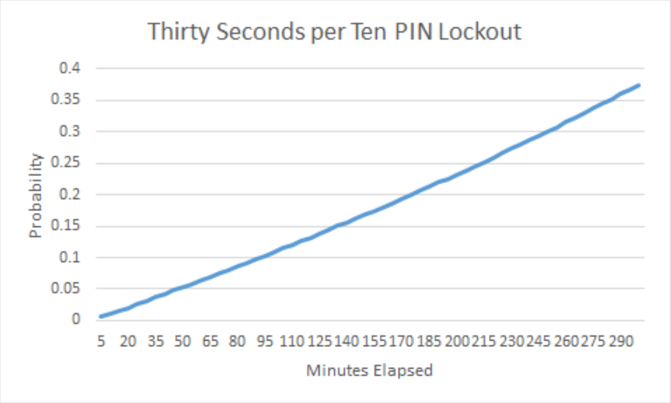

Figure 8. Thirty Seconds per Ten PIN Probabilities

$$P = \begin{bmatrix} .58 & .42 & 0 \\ 0 & .16 & .84 \\ 0 & 0 & 0 \end{bmatrix}$$

Given these results, the lockout period is not as effective as the per PIN delay. After five hours, we had a 37.5% overall success rate with a 42% chance of success to recover the first half of the PIN and an 84% chance to recover the second half of the PIN, an increase in probability across all statistics.

Next, we examined the same scenario except with different values for *s* and *p*. We increased *s* to 60 and decreased *p* to 5:

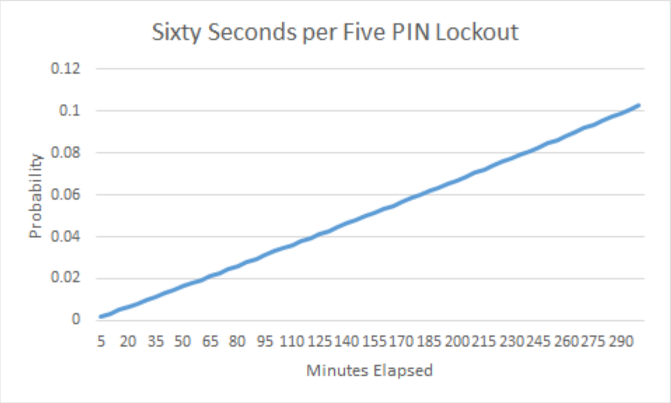

Figure 9. Sixty Seconds per Five PIN Probabilities

$$P = \begin{bmatrix} .89 & .11 & 0 \\ 0 & .95 & .05 \\ 0 & 0 & 0 \end{bmatrix}$$

The results of this simulation produced the lowest probabilities so far across the board. After 2.6 hours, we had only a 5% chance of success and an 8% chance after 4 hours. The success rate of the first PIN half was now 11%, whereas the success rate of the second half was only 5%. If we extended these results, we could formulate that an attacker could still attempt every PIN possible within 27 hours.

To further explore these results, we then combined both a per PIN delay and a PIN lockout period. Using the previous *s* and *p* values (s=60 and p=5), we added a *d* value of 5 and graphed the results:

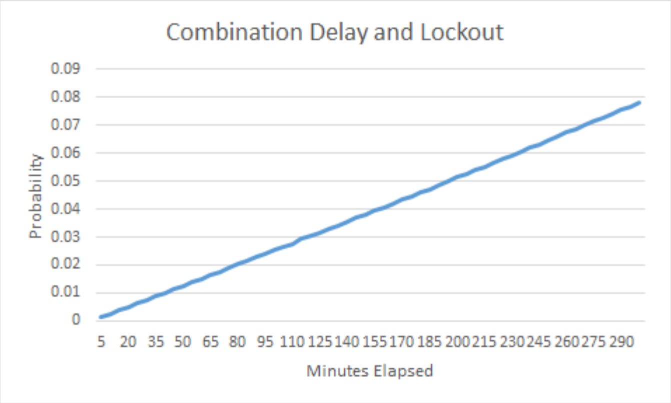

Figure 10. Combination Delay and Lockout Probabilities

$$P = \begin{bmatrix} .92 & .08 & 0 \\ 0 & .97 & .03 \\ 0 & 0 & 0 \end{bmatrix}$$

As expected, the probabilities had once again been lowered. The overall success rate was now 7% with a first half success rate of 8% and a second half success rate of 3%. Our overall success rate was 3% at 2 hours and 5% at 3.25 hours. Projecting these numbers further, we could see that an attacker could still try all possible PINs within 38.5 hours.

To make this data easier to analyze and compare, we then graphed all probabilities together. This helped put all of our data into perspective since every test shared a common time limit.

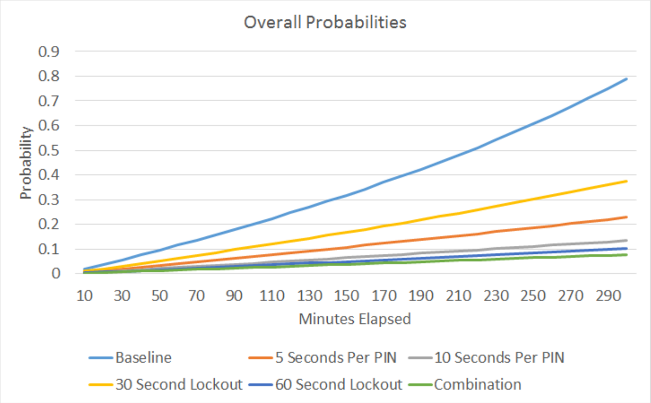

Figure 11. All Tested Probabilities

Based on the gathered data, we could see that these mitigations do help in lowering the probability of a successful WPS PIN brute force attack. However, none of the above examples will completely stop a dedicated attacker for obtaining the WPS PIN and subsequent WPA passphrase. Regardless of the mitigation used, there is a finite time that exists in which an attack can be 100% successful. These times for each scenario are show below for comparison.

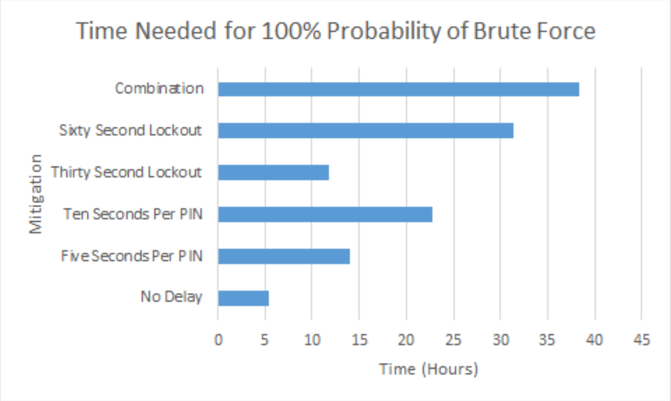

Figure 12. Time Needed for 100% Probability of Brute Force

This led us to believe that an alternate solution needed to be found for this problem. Many consumers and small businesses alike use routers with WPS capability to ensure the confidentiality and integrity of their transmitted data. Since disabling WPS on certain wireless access points is not an option, the device owners are stuck with permanently vulnerable devices. This is why I propose an alternate method to prevent against these types of attacks if the WPS standard is continually used in wireless devices.

One simple solution for this problem would be PIN randomization after a certain number of failed PIN attempts. This would keep the convenience features of WPS without sacrificing security. PIN randomization could be done via a cryptographically secure random number generator (CSRNG) after multiple failed attempts. This process would generate minimal overhead for devices that do not have extra processing power available. A more secure solution would additionally involve verifying all eight digits of the PIN at once instead of splitting the PIN into two different halves.

VII. Concluding Remarks

In conclusion, we can see that these mitigations are ultimately putting a bandage on the massive wound that WPS has opened. If manufacturers have no other choice but to implement PIN delay or lockout mechanisms in an attempt to secure their WPS enabled devices, then the obvious choice is to enable both a delay and a lockout. These additions will only further delay an attack for a finite amount of time. Wireless access point manufacturers and the Wi-Fi Alliance both have much work to do before this protocol is up to the security standards of today's technological environment.

References

[1] S. Viehbock, "Brute forcing Wi-Fi Protected Setup", www.packetstormsecurity.com, pp.2-9, Dec. 2011.

[2] D. Sora, "Wi-Fi Protected Setup – Security Enhancement or Threat?", The 8th International Scientific Conference "Defense Resources Management in the 21st Century". Brasov, Romania. pp.1-5, Nov. 2013.

[3] Wi-Fi Alliance, "Wi-Fi Protected Setup Specification", http://gpl.back2roots.org/, pp.34, December 2006.