EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

Detection and analysis of the Chameleon WiFi access point virus

Jonny Milliken[1*], Valerio Selis[2] and Alan Marshall[2,3]

Abstract

This paper analyses and proposes a novel detection strategy for the 'Chameleon' WiFi AP-AP virus. Previous research has considered virus construction, likely virus behaviour and propagation methods. The research here describes development of an objective measure of virus success, the impact of product susceptibility, the acceleration of infection and the growth of the physical area covered by the virus. An important conclusion of this investigation is that the connectivity between devices in the victim population is a more significant influence on virus propagation than any other factor. The work then proposes and experimentally verifies the application of a detection method for the virus. This method utilises layer 2 management frame information which can detect the attack while maintaining user privacy and user confidentiality, a key requirement in many security solutions.

Keywords: WiFi; Security; Virus; Infection; Propagation; Detection; Layer 2

1 Introduction

The increased availability of WiFi has occurred in spite of well-documented security vulnerabilities [1], such as denial of service (DoS) and rogue access point (rogue AP)[a] attacks. The consequence of this is that as demand drives up the availability and use of WiFi, the geographical area that an attack can exploit increases exponentially. It is pertinent to note however that currently the largest barrier to eradicating the threats to users and owners of WiFi networks is system and device misconfiguration, rather than inherent technology flaws [2]. This is revealed in the continued use of open and wired equivalent privacy (WEP) encryption in home and enterprise environments [3].

The primary means of defence against rogue APs in many cases is deployment of an IDS[b] (intrusion detection system), which alerts a human operator to the presence of an attack. The typical IDS method of detecting rogue APs is to track the location of the device, usually using received signal strength indicator (RSSI) values. In order to evade this detection, the attacker can attempt to copy the expected RSSI values by either placing the rogue AP within similar radius to the detector as the victim or editing the RSSI output to match the victim's RSSI values. In this case the legitimate AP and the fraudulent AP are resident in the network at the same time, which provides a mixed set of normal and abnormal traffic for detectors. Separating these two traffic streams then becomes the challenge in rogue AP detection.

Tactics exist for defeating rogue APs which masquerade either AP location or credentials. However, if the legitimate AP is not turned on or not broadcasting, then there is no normal traffic to compare to. Consider an instance where a legitimate AP is taken down and then a false AP comes up, with neither existing at the same time. Due to the prevailing assumption that two devices exist at the same time in a rogue AP attack, the current IDS systems have not been designed to detect this type of attack where the legitimate AP is taken over by the attacker. Hence, it is unlikely that current systems are able to detect this type of attack.

A new form of compromised AP attack has been demonstrated and analysed in [4], called the 'Chameleon' attack, perpetrated by the Chameleon virus. This attack replaces the firmware of an existing AP and masquerades the outward facing credentials. Thus, all visible and physical attributes are copied and there is no significant change in traffic volume or location information. Hence, this attack is considered advanced and difficult to detect, as IDS rogue AP detection methods typically rely on a

* Correspondence: mmilliken02@qub.ac.uk
[1]Institute of Electronics, Communications and Information Technology (ECIT), Queen's University Belfast, Belfast BT7 1NN, UK
Full list of author information is available at the end of the article

change in credentials, location or traffic levels. This work provides analysis of the Chameleon virus and demonstrates a method of detecting the propagation of the virus, as it constitutes an advanced rogue AP.

1.1 Background

Some works in the area [5-7] have considered a theoretical virus which has not been practically demonstrated. In [5] the authors analyse the spread of a theoretical virus via simulation. Results suggest that infection of a small number of routers can lead to thousands of infected devices within 24 h. Further work assesses the likely times users will be busy to avoid detection [6], showing that evading detection is a key principle of a WiFi virus. The work in [7] outlines the potential threats of a WiFi virus as traffic eavesdropping, destruction of traffic, disruption of infected host normal operations or even killing the device itself. As attacks and defences in network security tend to evolve together, the first step in devising effective countermeasures is to anticipate the malware hazards that could present themselves [7]. WiFi routers in particular have been identified as valuable targets for attack for malware, as no software currently exists to protect devices from such an infection [5], unlike in modern PCs where antivirus programs are well utilised.

Research in has investigated the behaviour of a practically demonstrated malware virus called Chameleon propagating over WiFi networks. This addressed aspects of virus propagation, structure, modelling and limits of successive propagation. The results show that the Chameleon virus can self-propagate via WiFi from AP-AP in a laboratory environment, but no method for successfully detecting the attack was available at that time. Although propagation was much less rapid than the results of simulations [5], the virus demonstrably evaded detection by a widely available open source IDS, constituting a zero-day attack.

This virus propagation can be considered an advanced form of rogue AP attack, which propagates directly between WiFi AP-AP embedded devices. Research into detecting rogue AP attacks is currently lacking [8], although they are considered a more technically difficult but growing threat in [9]. It is even estimated that as many as 20% of all enterprise locations already have a rogue AP connected to their systems and hence are particularly at risk from these attacks [8,10]. Users of WLANs in public environments are also considered vulnerable [11].

One of the most common metrics used to detect the presence of rogue APs in modern research is packet RSSI. The authors in [12] and [13] suggest that using a distributed set of sensors' sufficient RSSI information can be gathered to provide identification. This relies on different physical locations allowing for variations in traffic patterns; however, this is only applied to clients and not APs.

There is disagreement on the usefulness of RSSI in practical experiments. The authors in [13] and [10] conclude that use of RSSI as a WLAN location indicator is flawed as multipath effects and AP-specific processing of RSSI frame values severely impact results and make them unreliable. Furthermore, in [14] it is suggested that attackers, knowing RSSI is a detection metric, can alter their transmission power in frequent intervals in order to defeat the detection algorithm. Thus, the usefulness of RSSI as a metric for identification of location in rogue AP detection algorithms is uncertain.

The authors have previously shown that device identity can be attributed using layer 2 management frame information, rather than RSSI [15]. This method utilises packet inter-arrival times for beacon and probe frame exchanges to distinguish WiFi AP traffic. Developing a rogue AP detection method based on the prescribed identity would present a much more advanced and robust system for detection of rogue APs than current RSSI approaches.

This research extends the work undertaken in and [15] to identify new insights into the application, threat and detection of the Chameleon AP virus. Authors in outline the practical operation of the virus and show that a typical IDS cannot detect the attack. However, it does not consider in-depth analysis into the infectiousness of the virus or the impact of changing environmental variables and conditions, such as change in device susceptibility. Furthermore, it does not address a method of assessing the performance of such a WiFi virus. The work in [15] establishes that it is theoretically possible to detect rogue AP attacks using layer 2 management frames. It does not consider, however, the practical application against an attack. As the Chameleon virus represents an advanced rogue AP attack, evaluation of the layer 2 detection method against this threat demonstrates its effectiveness when practically implemented. This work presents two contributions: furthering the analysis of the propagation of the Chameleon virus and demonstrating the rogue AP detection ability of layer 2 traffic metrics through detection of the virus.

2 The Chameleon attack

The Chameleon virus employs a WLAN attack technique which independently infects and propagates amongst WiFi AP embedded systems. The propagation of the virus effectively constitutes an advanced rogue AP attack which is unique in that it occupies the exact location of the victim device. This contrasts with an evil twin attack, where an additional device appears which masquerades the credentials and broadcast information of the victim. In this attack the traffic volume significantly increases and conflicting RSSI measurements are observed for packets, which can be used by an IDS for detection purposes. As the Chameleon attack commandeers the existing hardware, it does not impact on either of these metrics. The principal steps of the Chameleon virus are as follows:

1. Establish a list of susceptible APs within the current location.
2. Bypass any encryption security on the AP.
3. Bypass the administrative interface on the AP.
4. Identify and store AP system settings.
5. Replace the AP firmware on vulnerable APs with the virus-loaded firmware.
6. Reload the victim AP system settings.
7. Propagate virus (return to 1).

Propagation of a virus over the wireless interface, rather than via wired backbone, presents several benefits for propagation. Firstly, the propagation of the virus would continue despite upgrades to backbone internet virus security methods, such as deep packet inspection. Without significant changes to AP operations, the attack can only be detected using WiFi frames. Secondly, the presence and volume of infected devices could not be assessed using existing forensic methods such as connection to suspicious IPs or known malicious domains. Thirdly, the virus can infect nodes which are not connected to the backbone internet. Finally, this virus is uniquely able to target APs within a specific region as propagation is based on proximity. Each of these factors presents additional challenges with detection of the virus if it is restricted to propagating over the wireless medium.

2.1 Infection model and epidemiology

The virus has been designed and practically demonstrated in a laboratory setting. This information was then used to inform an infection model to test the application of the virus in two urban environments: Belfast, Northern Ireland and London, England, with data extracted from Wigle.net. The region of Belfast under analysis contains approximately 14,553 APs, of which 22% are open, 61% are WiFi protected access (WPA) 2-encrypted and 14% are WEP-encrypted. The section of London under analysis contains approximately 96,433 APs, of which 24% are open, 48% are WPA/WPA 2-encrypted and 19% are WEP.

APs are considered to be connectable if their separation lies within a certain radius, varied between 10 and 50 m in the model. The model initiates the virus by infecting an AP at random to act as a seed and then calculates how many days would be required to either infect or blacklist all APs in the area. The blacklist contains all APs in the population for which virus infection has been unsuccessful, excluding them from subsequent attacks from other infected APs.

For every 2 days where there is no activity, the model randomly selects another reseed point. If the virus is able to self-propagate, then reseeding is not performed; however, if the virus cannot continue to propagate then it is restarted on another randomly chosen node to simulate the attacker instigating the virus again. The goal of this mechanism is to maintain a reseeding value as low as possible for continual propagation of the virus and assess how reliant it is on a human intervention. So reseeding is non-zero where virus propagation activity has halted, but returns to zero for every day where propagation is ongoing. Thus, we would expect that a lower reseed value implies a more potent virus, while a higher value indicates that the virus is terminating much more frequently and requires human intervention to be restarted. Each infected node attempts to infect all connectable APs within the vicinity for that day. When all APs have been processed, the infected node must wait until the following day to operate again. After 1 day of non-operation the node is blacklisted. A summary of the model results is presented in Table 1.

2.2 Virus analysis metrics

The difference in values for almost every comparison between Belfast and London and for each discrete value of radius within and between locations is shown in Table 1. The geographical density of APs is calculated as 454 APs/km^2 for Belfast and 3,013 APs/km^2 for London; however, these figures represent the *density of physical APs*. *Connectivity density* on the other hand rises as radius increases, as shown by rise in AP-AP connections in Table 1. As the radius grows, more potential victims are available and the observable density of victim APs from the viewpoint of the attacker rises. It is proposed here that the determining factor in this difference in radius and location results is primarily attributed to connectivity density. This impact calls into question the use of any one of the factors in Table 1 as an objective measure of successful WiFi virus propagation.

Analysis of the results in Table 1 indicates that the two important measures that have previously been used to quantify the virus spread are in fact not the most suitable criteria. The first metric that has been used is the time taken to infect a population (*days to completion*). Table 1 shows that for propagation of a practically constrained virus, the time becomes impractically large and is highly dependent on the connection radius of the APs. In many cases the time frame for selecting and infecting or blacklisting all of the nodes is in the region of tens of years and exhibits a range between locations

Table 1 Results of APV propagation depending on connection radius of AP for Belfast, Northern Ireland (NI) and London, England

	Virus model results for Belfast, NI				Virus model results for London, England			
Radii (m)	Days to completion	Total AP-AP connections	Total infected (*n* (%))	Reseeds	Days to completion	Total AP-AP connections	Total infected (*n* (%))	Reseeds
10	6,281	44,769	639 (4.5)	3,014	30,890	2,611,799	5,437 (5.6)	14,039
20	4,868	79,526	736 (5.1)	2,247	15,936	3,537,234	6,236 (6.5)	6,466
30	3,779	117,204	835 (5.7)	1,687	9,607	459,120	6,484 (6.7)	3,461
40	2,973	160,241	848 (5.8)	1,289	6,306	5,779,016	6,589 (6.8)	2,051
50	2,499	212,602	890 (6.1)	988	4,288	7,039,544	6,637 (6.9)	1,294

and radii of between 6.8 and 84.6 years. The same large deviation in time frames is observed for proportions of the total population, for example after 400 nodes infected, which is discussed in more detail in Section 2.4. Dependence on the time to infect a fixed number of nodes, whether that is the total population or a proportion of it, produces too much reliance on density to be indicative of objective virus performance. Although propagation is slow (and will also be addressed in Section 2.4), it constitutes a zero-day attack and presents an attractive attack vector for potential attackers. Nonetheless, this huge difference in the total infection time frames implies that it is a poor objective measure of WiFi virus performance as it is significantly affected by radius and density.

A second commonly used metric is the total number of infected APs in the population, signified as 'Total infected' in Table 1 (note that this value *excludes* APs infected solely for the purposes of reseeding). This number is demonstrably low for Belfast and London, approximately 5% and 6% of the total, respectively. While there would appear to be some similarity in the percentage infected across radii (e.g. 4.5% to 6.1% for the Belfast trial), the variance between both sets of trials (i.e. Belfast 4.5% for 10 m and London 6.9% for 50 m) is relatively large, implying that the increase in propagation performance is more related to the AP density than the infectiousness of the virus. Infectiousness here refers to the capability of the virus to automatically propagate itself amongst vulnerable hosts. In practice this quantity is highly dependent on density, i.e., the denser the APs, the more APs will become infected and the better the virus may appear to perform, even though the virus behaviour has not been improved. This makes it very difficult to differentiate the performance of the virus from the environment over which it operates. Therefore, the total number of infected APs by itself is not a clear indicator of how well the virus can spread.

As shown in Figures 1 and 2, the percentage increase in Total infected also compares unfavourably with the *reseed* value, as an almost 10-fold decrease in reseeds produces a mere 10% increase in Total Infected. A reduction in the number of reseeds would imply that the virus is more independent of human intervention and should be utilising node infections to propagate. This should equate to a proportionally larger number in Total Infected, but this is not the behaviour observed. A high reseeding value is a strong indicator that the ability of the virus to successfully propagate without constant human intervention is poor, theoretically making the impact of the virus less severe. The reseed values also exhibit a large variance across radii and locations, excluding it as a measurement.

The effect of the increase in AP-to-AP connections is not as would be expected, since in both trials this increase is much higher than the increase in Total infected. The reasons for this result are unclear, although it may be attributable to the effect of diminishing returns; for example, the trebling of the number of connections between radius 10 m and radius 50 m for London may be curtailed by the low infection rate of the virus. Clearly, none of the available traits alone in Table 1 are suitable. A more effective measure of viral performance is needed.

2.3 Independence ratio

In search of an appropriate objective measure of viral performance, multiple combinations of the metrics in Table 1 were considered. The combination which

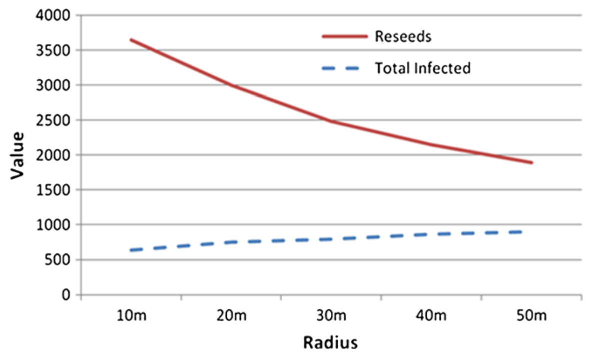


Figure 1 WiFi virus reseeding value versus total infected nodes for all radii (Belfast).

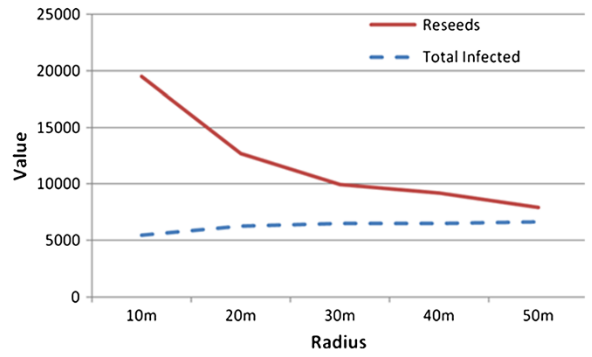


Figure 2 WiFi virus reseeding value versus total infected nodes for all radii (London).

presented the largest stability across radii and density is the ratio between days to completion and reseed value defined here as the independence ratio:

$$\mathrm{Ratio}_{\mathrm{Ind}} = \text{Days to completion}/\text{Reseeds} \qquad (1)$$

As seen in Figures 3 and 4, this ratio exhibits a similar range across all measurement iterations, falling between 2.1 and 2.5 in Belfast and between 2.1 and 3.3 for London. The ratio expresses the reliance of the virus on reseeds for infection speed, which takes into account several propagation factors. For example, were days to completion to remain constant, if the reseed value falls then a higher independence ratio results, indicating that less human interaction is required to propagate the virus. As the reseed value rises, then the ratio reduces, indicating poorer performance and reduced threat.

Alternatively, if reseeding remains constant, then an increase in days to completion increases the independence ratio and the virus can be considered less reliant on human intervention to propagate. If the days to completion reduce relative to the reseed value, then the virus is less independent and thus there is less implied threat. While this may appear to be counter-intuitive, the reason that an increase in days to completion is not considered an indicator of poor virus propagation is only under the condition that it increases *relative* to the reseed value.

One of the most important factors for the continued success of the malware is the ability to evade detection. For this reason the virus does not attempt to propagate if a client is currently connected, which would appear to waste time. However, secrecy is considered more important than speedy propagation. Thus, an increase in days to completion is not a negative occurrence if other factors are taken into account. For a set reseed value, an increase in time means that the virus has been able to survive without being detected for a longer period without human intervention.

Note that an increase in days to completion is a positive attribute only if this is achieved with respect to reseeds. It is still the case that waiting 6 to 85 years, as in Table 1, is disproportionately long to be used in itself as an indicator and that an increase in this value in isolation implies poorer performance.

2.4 Infection acceleration

For infection of the entire population, there is a large variance in time frames which does not increase linearly, as in Figures 5 and 6. Note that for a subset of the total population infected, after 400 infected nodes for example, there is a large deviation (approximately 700%) in the total time required to infect between the maximum and minimum radii for Belfast. This deviation then increases further between Belfast and London. This provides further evidence that using an absolute value of infected nodes, or a value proportional to the total population, is not an appropriate virus analysis metric as it is significantly impacted by density.

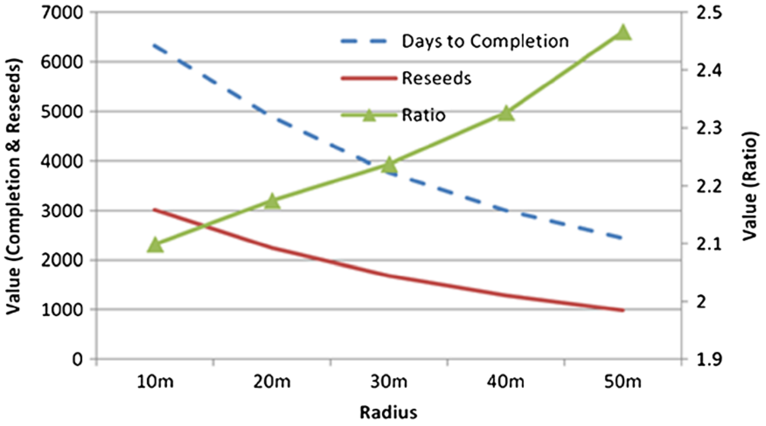


Figure 3 Independence ratio between completion time and reseeding value for Belfast.

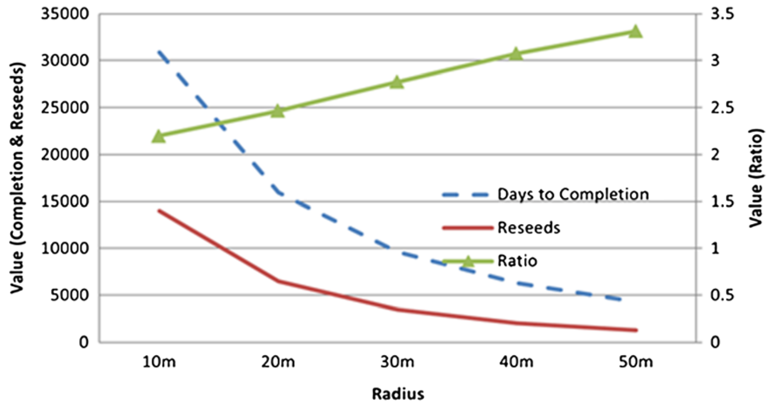


Figure 4 Independence ratio between completion time and reseeding value for London.

As the population is either blacklisted or infected, the rate of activity slows down. The switching point between acceleration and deceleration can be determined by analysing the gradient of the curves in the figures. Once the gradient tends towards flattening out, the acceleration is deemed to have stopped. The first 5 years are recognised as exhibiting the most successful infection activities. In this sense the time for infection behaviour of the virus is analogous to the principle of a 'biological half-life'.

Focusing on the lowest density curve, in radius of 10 m for Belfast, after 5 years the total infected value is greater than 50% of maximum, after which the virus requires another 12.3 years to assess (i.e. infect or blacklist) the final 50%. For London approximately 17 years are required for the number of the total infected to pass 50% and the virus requires another 67.6 years to infect the remainder.

For a radius of 50 m after only 3 years, the number of totally infected is greater than 70% in both locations. In particular, for Belfast, the virus needs another 3.7 years to assess the last 30% in order to complete the propagation, while for London another 8.7 years is needed to assess the final 30%.

This effect is explained as an indication of the isolation of the APs in each location. The virus accelerates in a totally virgin environment but decelerates as other infected or blacklisted nodes present barriers. The conclusion drawn is that the risk of an epidemic occurs much earlier than as would otherwise be indicated by the days to completion results.

2.5 Impact of manufacturer susceptibility

The next investigation of viral performance concerns quantifying the impact of increasing AP product susceptibility. The value of product susceptibility expresses how many APs in the vicinity are vulnerable to the virus, as governed by Equation 2.

$$P_{\mathrm{Inf}} = P_{\mathrm{Prod\_sus}} X \left(1 - P_{\mathrm{Enc\_nosus}}\right) X \, P_{\mathrm{Admin\_sus}} \qquad (2)$$

where $P_{\mathrm{Prod\_sus}}$ is the probability of the product being susceptible to the attack, $P_{\mathrm{Enc\_nosus}}$ is the probability of the encryption key being resistant to bypassing (0% if not present) and $P_{\mathrm{Admin\_sus}}$ is the probability that the admin credentials are susceptible to a word combination in a 1-M word dictionary.

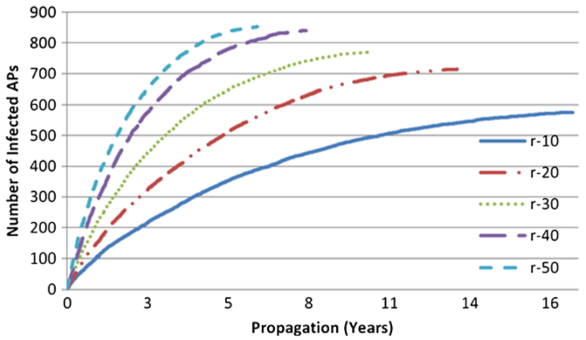


Figure 5 Infected node population (Belfast).

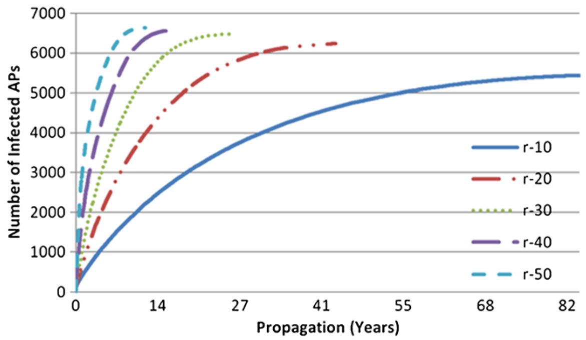


Figure 6 Infected node population (London).

This can also be considered an investigation into the effect of device heterogeneity in the population on viral propagation, as higher susceptibility is analogous to lower heterogeneity. In order to assess this effect within the population, the proportion of products susceptible to the virus in Equation 2 ($P_{\mathrm{Prod\_sus}}$) has been varied from 10%, the value used in all previous calculations, to 60%. Curves for susceptibility values outside of this range can be extrapolated from the data. This effect produces a proportional change in the total number of infected devices, as seen in Figures 7 and 8; however, this still relates to a small percentage of the total available APs.

Restrictions on the influence of $P_{\mathrm{Prod\_sus}}$ may be caused by the relative distance between APs and the difficulty in connecting between them over a large area. Thus, the susceptible APs are still sufficiently separated and at the mercy of geography such that increasing the vulnerable firmwares has limited influence on performance. We deduce from this result that AP connectivity density is a more significant factor for virus propagation than product susceptibility.

2.6 Spread of infected area

The current implementation assumes that AP-AP infection is the only viable route and does not take into account the possibility of clients becoming infected. Breaching the AP-client link is the only major barrier to true virus mobility. While this aspect has not been considered here, possibilities may be presented through phishing or URL redirection techniques. Should this hurdle be surmounted, then the current restriction to virus propagation (i.e. the AP connection radii) is removed and the attack becomes much more dangerous. The attack area would no longer be fixed, and the virus carrying clients (virus carriers) could spread the virus amongst APs they come into contact with.

Considering this, a major factor in the spread of the virus may not be the number of APs infected, but the physical area over which these infected APs can connect to clients. Thus, any client which resides even temporarily within this infected area and connects to the infected AP could find itself susceptible to infection. Figures 9 and 10 show the spread of infected area, as increasing numbers of APs are infected for each radius and site.

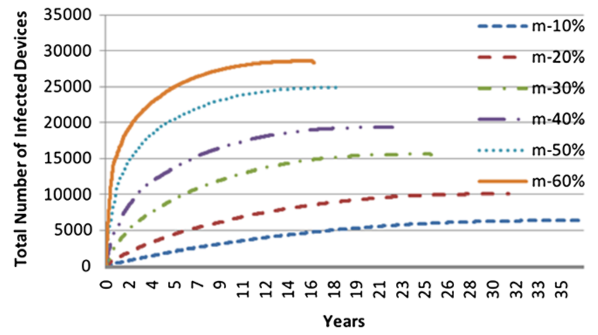


Figure 8 Effect of product susceptibility on infection rate (London, radius = 30).

The maximum limit of infected area is given as approximately 9 km^2 for a radius of 50 m in Belfast and approximately 72 km^2 for 50 m radius in London. This equates to 25% and 225% of the total area under consideration, respectively. For Belfast this is a significant portion of the area under consideration, and a large geographical area, which allows APs in the vicinity to connect to a large volume of mobile clients. For London the value of 225% indicates that over twice the geographical area is covered by the APs. This result is a consequence of the method used to generate overlap information.

The area is calculated in the model as the total area covered by two overlapping radii, the attacking device and the victim. This process does not take into account overlaps of more than two APs, as this was deemed to add too much complexity. In locations of sparse infection and connectivity, this is deemed to be an acceptable estimate, as the likelihood of a significant amount of multiple infected APs covering the same area is remote.

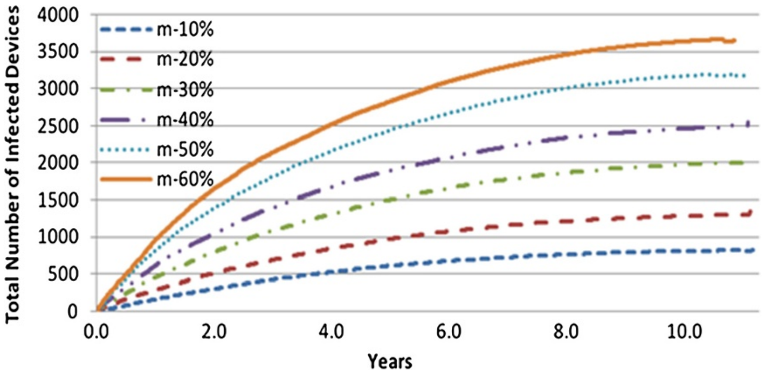


Figure 7 Effect of product susceptibility on infection rate (Belfast, radius = 30).

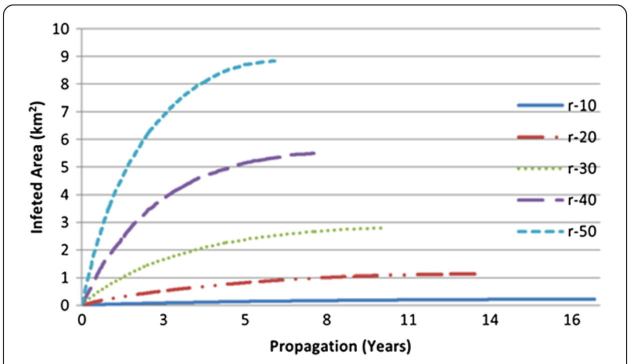


Figure 9 Spread of infected area due to AP radius (Belfast).

This is not necessarily the case for high-density environments, as indicated by the 225% infection area coverage attributed to London. What this result does show, however, is that for London the area is sufficiently dense that multiple APs are covering the same geographical area, so a smaller percentage of APs are required to infect the maximum coverage area, showing that connectivity density is a determining factor in the success of the virus.

3 Attack detection

The Chameleon attack is a significant threat to WiFi security with implications including data theft and device malfunction. It has been shown in [4] that the current layer 2 wireless intrusion detection systems (WIDS), Kismet, has a low probability of detecting this attack or the propagation of the virus through an environment. This is principally due to the lack of both attacker and victim traffic existing on the network at the same time. The result is a highly dangerous and pervasive WLAN threat that has not been defended against in previous work.

Due to the nature of the attack, it can only be tested in a laboratory environment, outlined in Figure 11, rather than in a live testing environment. During the testing carried out in [4], layer 2 wireless traffic was collected by the Kismet IDS which failed to detect the presence of the attack. This same data can be employed here, through offline processing, to design a method of detecting this attack.

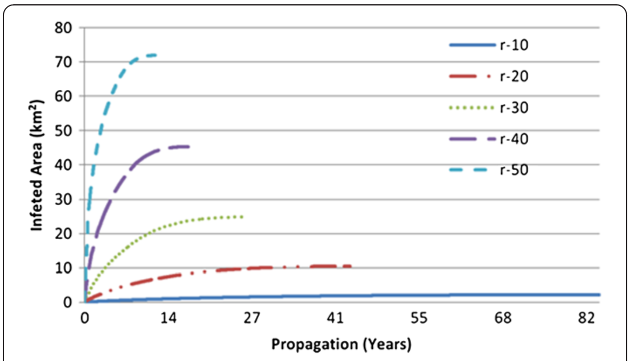


Figure 10 Spread of infected area due to AP radius (London).

The use of layer 2 management frame traffic from WiFi networks has many positive attributes for research applications. As this management header traffic is broadcast in plaintext and devoid of any encryption, it can be collected with fewer privacy or confidentiality concerns. These concerns are often a major barrier to performing practical WiFi network research in live environments.

The distributions of beacon inter-arrival times before and after an attack indicate that a subtle, potentially detectable change has occurred. Figures 12 and 13 show the difference in the pre-attack and post-attack distributions for APV-P1 in Figure 11. Comparison of these graphs demonstrates that while the modal peak remains in both distributions, the distribution of higher interval values are different. The modal peak, which resides around 0.1024 s, represents the ability of the device to produce beacons at a rate of 10/s. The higher interval values are expected to represent firmware and device operational nuances (e.g. frame processing or transceiver response time), so the change in these ranges represents the best detection potential for a device identity system.

It is important to note that none of the APs under consideration here was in operational use and thus have no uncontrolled traffic on them other than that produced by the client generators indicated as 'C' in Figure 11. Thus, deviations can be said to be caused entirely by external entities such as people, networks and network behaviour. In the presence of these interference sources, the perpetration of the attack can be observed visually. These findings are typical of the majority of instances where the attack was observed in the laboratory tests.

3.1 Detection method

Figures 14 and 15 demonstrate the beacon intervals over the course of a Chameleon infection of an AP, as observed from the IDS location in Figure 11. It can be seen in Figure 14 that the attack clearly changes the interval values; however, the change is less evident in Figure 15. It is difficult to determine where the attack has taken place, although an algorithmic approach could improve this.

The overlay of an exponentially weighted mean (running average) on the dataset produces a good indicator of the expected average values over time, as demonstrated in Figures 14 and 15. This average is fairly resistant to large deviations in the data; however, it is too insensitive to significant changes in average deviation to be useful as a detection technique by itself. Figures 14 and 15 also contain an overlay of a windowed average

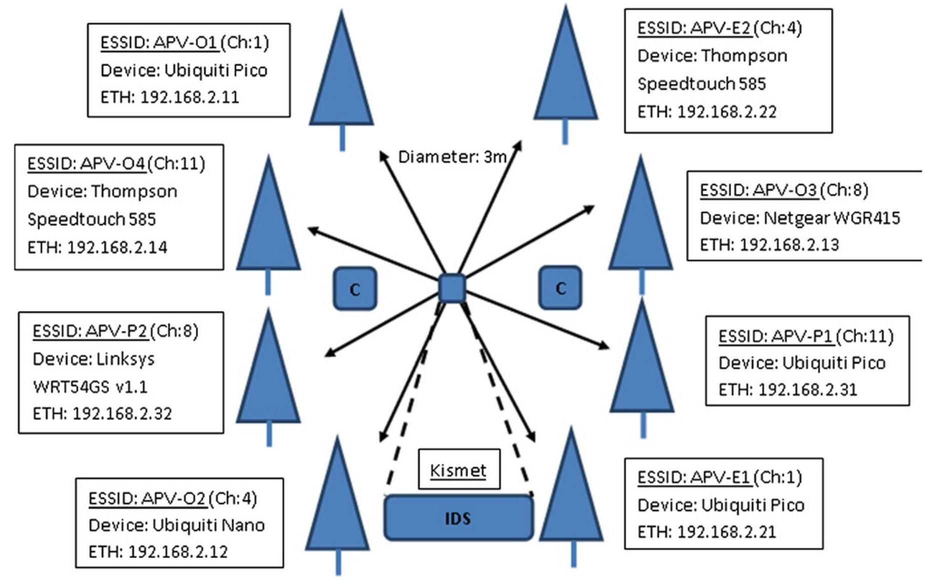


Figure 11 Laboratory setup.

(size 20 frames). In each of these cases the windowed average is sensitive to the change in average interval where the running average is not. This indicates that a combination of the two averages can be employed to act as the identification algorithm.

The addition of a windowed average which is more sensitive to change raises the question of outlier removal. A strict and adaptive outlier detection method ensures dynamic selection of outliers based on current averages. This dynamic outlier threshold algorithm operates as follows, where the algorithm is independently applied for windowed and running averages:

1. Wait until settling has been observed for the average following the method presented [15].
2. Create outlier detection bounds above and below the average based on the standard deviation multiplied by an outlier standard deviation range (initialised as 4).
3. If an outlier is detected as lying outside of these bounds it is excluded from further average calculations, but the number of outliers is incremented in a variable.
4. If the number of outliers detected represents larger than 0.03% of the total observations, then the outlier standard deviation range is increased by 3% of the current value.
5. If an observation is not detected as an outlier, then the outlier standard deviation range is decreased by 1% of the current value (the absolute limits of

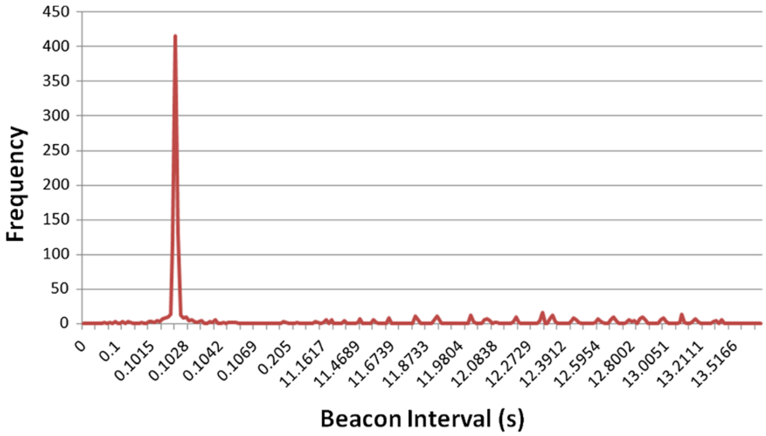


Figure 12 Pre-attack distribution of beacon intervals.

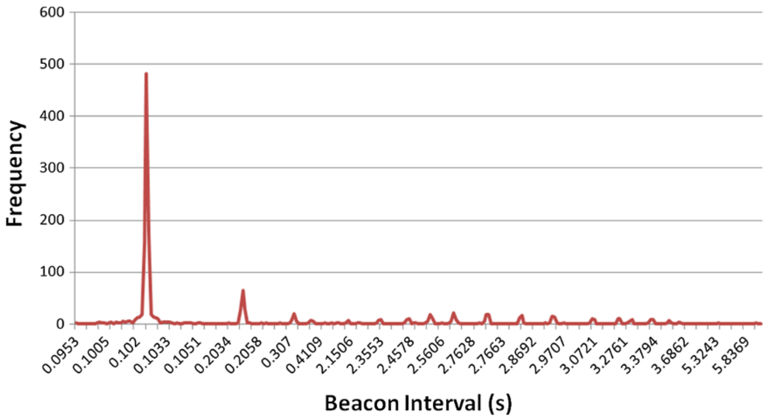


Figure 13 **Post-attack distribution of beacon intervals.**

outlier standard deviation bound are restricted to between 3 and 5).

This process is described as pseudo-code in Figure 16. This outlier detection system has been employed to remove impulse outliers, i.e. those which are created by temporary interferers rather than a general change in the trend of the traffic. The bound increases slowly once outliers are detected but will return to the original value if the outliers then disappear. However, if many outliers are detected, then they may be considered a permanent change in the average value and are then allowed to influence the average as the bound increases to include them.

The choice of increase percentage, decrease percentage and absolute limits is based on the expected level of outliers in a standard normal deviation. In the case of a normal distribution, 99.7% of all traffic should be found within three standard deviation bounds. Thus, outliers should make up 0.3% of the traffic. Should the outliers be deemed to represent more than 0.3% of the total traffic, then the algorithm considers that there may be a change in the average and increases the outlier bounds to accommodate this new traffic. If this new traffic continues to exhibit values outside of the bounds, then the average changes accordingly (3%) and the standard deviation bound can reduce again once the new average has been altered; however, it reduces more slowly (1%) since values within the bounds are more common. The bounds are restricted to between 3 and 5 to enforce stability and still exclude major outliers, while taking into account the possibility of slow change to the average.

This outlier removal algorithm excludes many of the impulse outliers and allows the remaining data to be used to estimate suspicious changes in the interval average value. The algorithm used to detect this illegitimate change operates as follows:

1. Wait until settling has been observed for the beacon frames, following the method in [15].

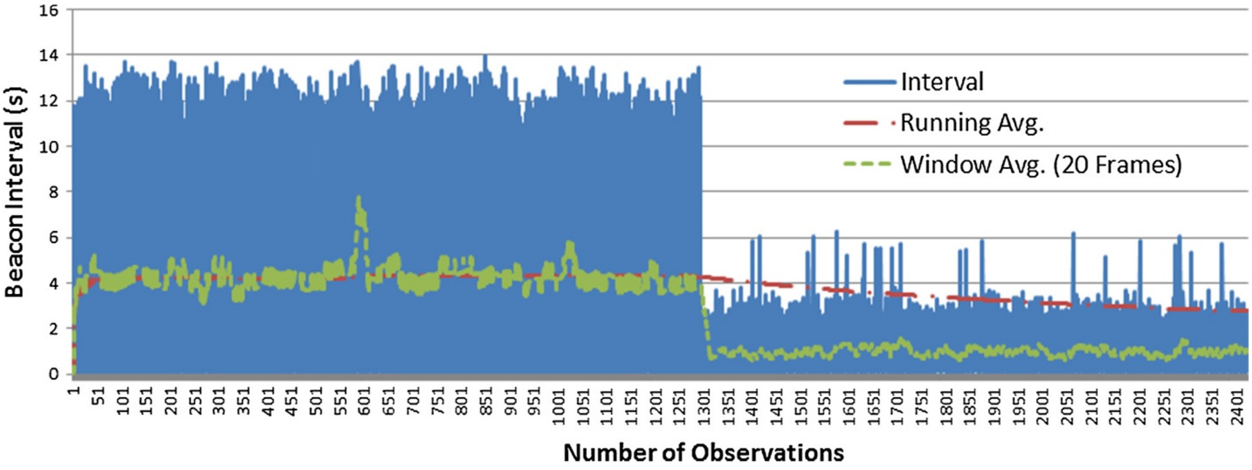


Figure 14 **Demonstration of the difference between running average and windowed average (20 frames) for APV-P1.**

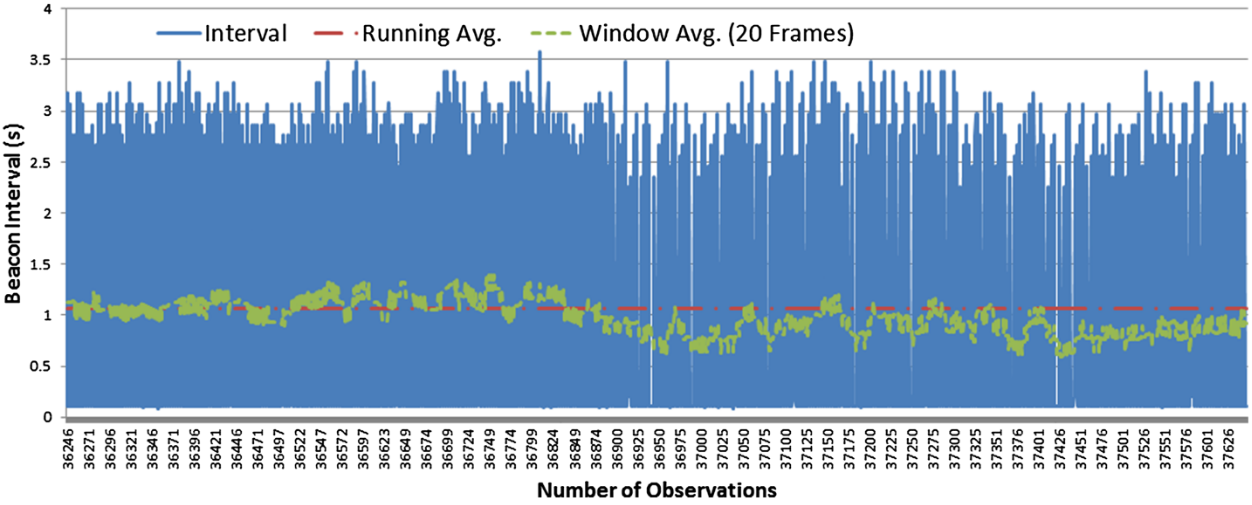
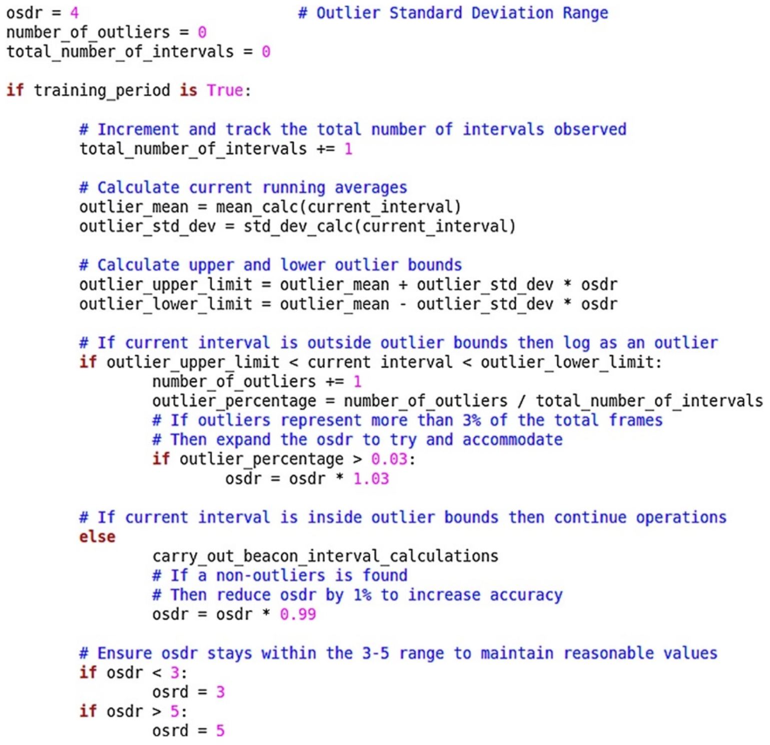


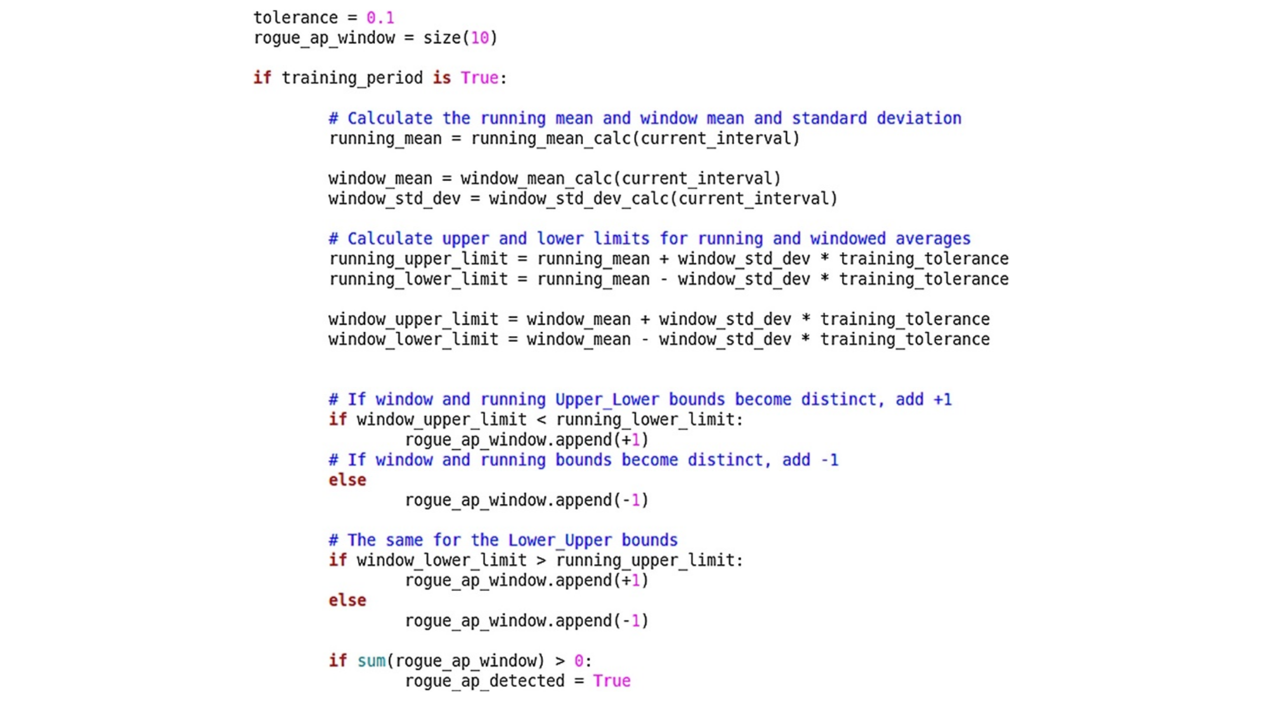
Figure 15 Demonstration of the difference between running average and windowed average (20 frames) for APV-E1.

2. Set upper and lower bounds to 10% of the windowed standard deviation for the running average and the windowed average, as presented in [15].
3. Compare the lower bound of the windowed average to the upper bound of the running average and vice versa.
4. If the windowed bounds violate either of these conditions, these are attributed a value of +1, otherwise attributed a value of −1.
5. Maintain a running sum of the previous 20 numerical values stated in step 4.
6. If this sum exceeds the 0 threshold point then the Chameleon attack has been detected.



```
osdr = 4                         # Outlier Standard Deviation Range
number_of_outliers = 0
total_number_of_intervals = 0

if training_period is True:

        # Increment and track the total number of intervals observed
        total_number_of_intervals += 1

        # Calculate current running averages
        outlier_mean = mean_calc(current_interval)
        outlier_std_dev = std_dev_calc(current_interval)

        # Calculate upper and lower outlier bounds
        outlier_upper_limit = outlier_mean + outlier_std_dev * osdr
        outlier_lower_limit = outlier_mean - outlier_std_dev * osdr

        # If current interval is outside outlier bounds then log as an outlier
        if outlier_upper_limit < current interval < outlier_lower_limit:
                number_of_outliers += 1
                outlier_percentage = number_of_outliers / total_number_of_intervals
                # If outliers represent more than 3% of the total frames
                # Then expand the osdr to try and accommodate
                if outlier_percentage > 0.03:
                        osdr = osdr * 1.03

        # If current interval is inside outlier bounds then continue operations
        else
                carry_out_beacon_interval_calculations
                # If a non-outliers is found
                # Then reduce osdr by 1% to increase accuracy
                osdr = osdr * 0.99

        # Ensure osdr stays within the 3-5 range to maintain reasonable values
        if osdr < 3:
                osrd = 3
        if osdr > 5:
                osrd = 5
```

Figure 16 Outlier detection pseudo-code.



```
tolerance = 0.1
rogue_ap_window = size(10)

if training_period is True:

        # Calculate the running mean and window mean and standard deviation
        running_mean = running_mean_calc(current_interval)

        window_mean = window_mean_calc(current_interval)
        window_std_dev = window_std_dev_calc(current_interval)

        # Calculate upper and lower limits for running and windowed averages
        running_upper_limit = running_mean + window_std_dev * training_tolerance
        running_lower_limit = running_mean - window_std_dev * training_tolerance

        window_upper_limit = window_mean + window_std_dev * training_tolerance
        window_lower_limit = window_mean - window_std_dev * training_tolerance


        # If window and running Upper_Lower bounds become distinct, add +1
        if window_upper_limit < running_lower_limit:
                rogue_ap_window.append(+1)
        # If window and running bounds become distinct, add -1
        else
                rogue_ap_window.append(-1)

        # The same for the Lower_Upper bounds
        if window_lower_limit > running_upper_limit:
                rogue_ap_window.append(+1)
        else
                rogue_ap_window.append(-1)

        if sum(rogue_ap_window) > 0:
                rogue_ap_detected = True
```

Figure 17 Attack detection pseudo-code.

This process is described as pseudo-code in Figure 17. The algorithm estimates how similar the current (windowed) average is to what the expected (running) average should be for a specific access point. If this windowed average is statistically different, based on the standard deviation, from the running average for a significant length of time (20 frames in this case), then an attack can be registered.

Figures 18 and 19 plot the behaviour of the beacon summation algorithm for APV-P1 and APV-E1. The dashed green line, entitled ‘SUM of 20 Previous deviations’, demonstrates the observation at which the attack was detected according to the attack detection pseudo-code. Once the 0 SUM threshold has been breached, the attack has been detected, regardless of whether the SUM subsequently drops below the threshold again. Both of these images indicate that this system is able to accurately detect the presence of the Chameleon rogue AP attack. The results from applying this algorithm to the test dataset are given in Table 2, which show that the attack can be detected

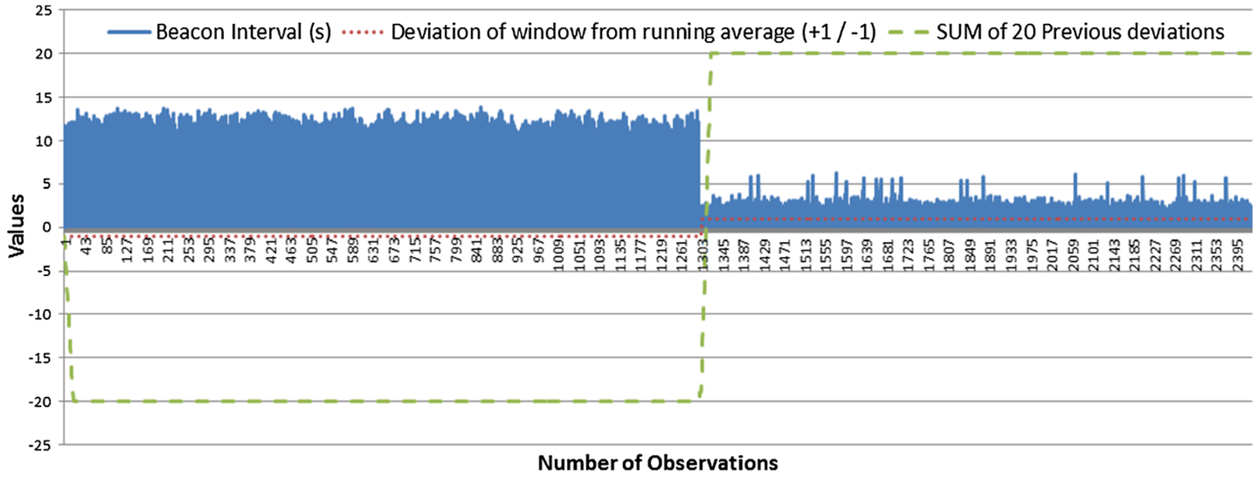


Figure 18 Performance of the summation algorithm for APV-P1.

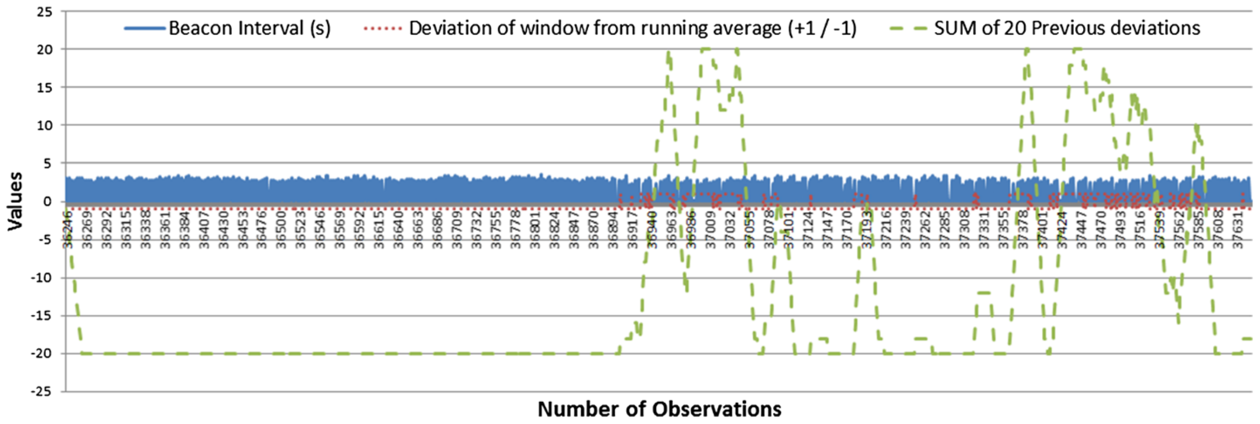


Figure 19 Performance of the summation algorithm for APV-E1.

between approximately 23 and 97 s after the completion of the attack. Within this time, approximately 30 to 60 beacon observations were required to come to this conclusion. These values represent low requirements on time and observations for attack detection.

4 Conclusions

To aid analysis of WiFi viral performance, this work has proposed a new metric for virus propagation performance. The results presented indicate that the use of total numbers infected, infectiousness, reseeding or infection time frame alone are poor barometers of success of this type of virus. This is due to the large influence of either location or connection radius size on all of the factors. Rather, it is shown that the ratio between the time frame for total infection (days to completion) and the level of reseed required (*independence ratio*) is a more objective measure of success, as it indicates the level of human interaction required in order to maintain a level of self-sustaining propagation.

This work demonstrates that AP connectivity density is a more important factor for virus propagation than product susceptibility. Furthermore, the virus accelerates in a totally virgin environment, but decelerates as nodes are infected, showing that the risk of an epidemic occurs much earlier than as would otherwise be indicated by the days to completion results. The geographical area which an infected network of APs can cover is estimated as falling between 25% and 225% of the total area under consideration. This shows that even for small numbers of infected APs, the coverage area of an urban area can be significant, presenting danger to clients in the vicinity who may connect to infected APs and indicating that an AP-client infection method could greatly increase the threat of the virus. Hence, connectivity density is currently the defining factor in success of the virus, but this may not be the case if AP-client infections are employed.

Table 2 Results of summation algorithm for each AP in the laboratory experiment

AP	Observed number of attack	Observed number of detection	Running average interval (s)	Time until detection
O1	1,091	1,122	3.13	97.03 s
O2	994	1,025	0.75	23.25 s
E1	36,876	36,942	1.13	74.58 s
P1	1,298	1,329	3.03	93.93 s
P2	16,899	N/A	0.75	N/A

The example locations chosen represent high- (London) and medium (Belfast)-sized urban environments, based on the density of APs. Density of APs then impacts quite significantly on the infection ability of the virus, or any computer virus which relies on physical proximity. While only two cities have been investigated, the effects presented here demonstrate that as population density increases, the effects and threat posed by such a virus increase.

Finally, this work presents a method of detecting the Chameleon virus using WiFi layer 2 packet average intervals for beacon frames. The virus constitutes an advanced rogue AP attack which copies both device credentials and location and does not require increased traffic levels to operate. This renders it outside of the detection ability of current detection systems and has until now not been shown to be experimentally detectable. Two algorithms have been developed to facilitate this: dynamic outlier detection and AP traffic identity detection. This can be used to attribute identity to a specific AP from that collection location. Application of this identity attribution system has been shown to detect the Chameleon attack in a laboratory environment within a time frame of 23 to 97 s or 30 to 60 observations of beacon intervals.

Endnotes

[a] http://www.giac.org/paper/gsec/4060/rogue-wireless-access-point-detection-remediation/106460

[b] http://www.sans.org/reading-room/whitepapers/detection/understanding-intrusion-detection-systems-337

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors gratefully acknowledge the assistance of EPRSC under grant number EP/H004793/1.

Author details

[1]Institute of Electronics, Communications and Information Technology (ECIT), Queen's University Belfast, Belfast BT7 1NN, UK. [2]Traffic Observation via Management (TOM LTD), Northern Ireland Science Park, Belfast BT7 1NN, UK. [3]Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK.

Received: 2 July 2013 Accepted: 3 October 2013
Published: 17 October 2013

References

1. CPP, *UK wireless network hijacking, a CPP white paper. (CPP, 2010)*. http://www.slideshare.net/CPPUK/uk-wireless-network-hijacking-2010. Accessed 25 Jun 2013
2. AirTight Networks, *Wireless (in)security trends in the enterprise. (Airtight Networks, 2009)*. http://go.airtightnetworks.com/Wireless-InSecurity-Trends-in-the-Enterprise.html. Accessed 25 Jun 2013
3. AirTight Networks, *Financial districts wireless vulnerability study. (Airtight Networks, 2009)*. http://www.airtightnetworks.com/fileadmin/pdf/resources/Financial_Districts_–_WiFi_Scan_-_Version_8.pdf. Accessed 25 Jun 2013
4. J Milliken, *An investigation into MAC-layer intrusion detection strategies for open-access wireless LANs, Thesis* (Queen's University, Belfast, 2012)
5. H Hu, S Myers, V Colizza, A Vespignani, WiFi networks and malware epidemiology. Proc Natl Acad Sci U S A **106**(5), 1318–1323 (2009)
6. N Husted, S Myers, *Why mobile-to-mobile wireless malware won't cause a storm* (Paper presented at the USENIX conference on large-scale exploits and emergent threats, Boston, USA, 2011)
7. HMR Khouzani, S Sarkar, E Altman, *Maximum damage malware attack in mobile wireless networks* (Paper presented at the international conference on computer communications, San Diego, USA, 2010)
8. G Shivaraj, M Song, S Shetty, *A hidden Markov model based approach to detect rogue access points* (Paper presented at the IEEE military communications conference, San Diego, USA, 2008)
9. NJ Percoco, *Trustwave global security report 2010 (Trustwave 2010)*. http://trustwave.com/downloads/whitepapers/Trustwave_WP_Global_Security_Report_2010.pdf. Accessed 27 Jun 2013
10. L Ma, AY Teymorian, X Cheng, *A hybrid rogue access point protection framework for commodity WiFi networks* (Paper presented at the IEEE 27th international conference on computer communications, Phoenix, USA, 2008)
11. S Nikbakhsh, ABA Manaf, M Zamani, M Janbeglou, *A novel approach for rogue access point detection on the client-side* (Paper presented at the 26th international conference on advanced information networking and applications workshops, Fukuoka, Japan, 2012)
12. Z Tao, B Nath, A Lonie, *X-mode: A real time approach of discriminating WiFi networking impersonators* (Paper presented at the4th international conference on next generation web services practices, Washington DC, USA, 2008)
13. DB Faria, DR Cheriton, *Detecting identity-based attacks in wireless networks using signalprints* (Paper presented at the ACM workshop on wireless security, Evanston, USA, 2006)
14. V Nagarajan, V Arasan, D Huang, *Using Power Hoping to Counter MAC Spoofing Attacks in WLAN* (Paper presented at the IEEE consumer communications and networking conference, Las Vegas, USA, 2010)
15. J Milliken, V Selis, KM Yap, *A Marshall, development of device identity using WiFi layer 2 management frames for combating rogue APs* (Paper presented at the international conference on security and cryptography, Reykjavik, Iceland, 2013)

doi:10.1186/1687-417X-2013-2
Cite this article as: Milliken *et al.*: Detection and analysis of the Chameleon WiFi access point virus. *EURASIP Journal on Information Security* 2013 **2013**:2.