



Orange Cyberdefense

Weaknesses in Secure Remote Access

Gaps left by VPN implementations leave remote users unprotected

Wicus Ross & Charl van der Walt

3 August 2020

@wicusross
@charlvdwalt
<https://orangecyberdefense.com/global/insecure-remote-access/>

wicus.ross@orangecyberdefense.com
charl.vanderwalt@orangecyberdefense.com

Version control

Version	Date	Change Description
1.0	03 August 2020	First public release

Editor / Approver	Function	Contact details
Wicus Ross	Senior Researcher	wicus.ross@orangecyberdefense.com
Charl van der Walt	Head of Security Research	charl.vanderwalt@orangecyberdefense.com

Not so Secure Remote Access

Contents

Background	6
A note on terminology.....	8
TLDR.....	8
Executive Summary of Findings	8
Standard Mode Test Outcome.....	9
Lockdown Mode Test Outcome.....	10
Executive Summary of Recommendations.....	10
The trade-offs we make and the reality we face.....	11
What is a VPN supposed to do?	13
How a VPN works and what it does.....	14
Technical Background	18
Introducing captive portals.....	18
Captive portal Behaviour	19
User Initiation.....	20
Captive portal Presentation	20
Access Restrictions	21
Online State Detection.....	22
Linux and captive portal.....	22
DNS Suffix Search List	22
Research Overview	23
Working Assumptions	23
Fundamental Research Question	24
Research Scope.....	24
Vendor Engagement.....	25
Test Scenarios	25

Captured State	26
Online State	26
VPN in Standard Mode	26
VPN in Lockdown Mode	27
Target Security Model	27
Technical Threat Scenarios	28
Research Methodology	31
Test Environment.....	32
Windows Client Environment.....	32
Windows Host Configuration.....	32
Captive portal.....	35
Attacker Platform	35
Test Cases.....	37
Test Case 1 – Sniffing Data	37
Test Passed:	37
Test Failed:	37
Test Case 2 – DNS ‘Person-in-the-Middle’ / Poisoning.....	38
Test Passed:	38
Test Failed:	39
Test Case 3 – Credential Harvesting	40
Test Passed:	41
Test Failed:	41
Additional Test Case Execution Notes:.....	41
Test Case 4 – Windows NTLM Hashes Theft with Responder.....	42
Test Passed:	43
Test Failed:	43
Test Case 5 – Browser Tunnelling with BeEF Hooks.....	44
Test Passed:	44
Test Failed:	44
Test Case 6 – Interact with host using IPv6.....	46
Research Findings	48
Standard Mode Test Outcome.....	48
Comment on Standard Mode Configuration	48
Lockdown Mode Test Outcome.....	49

Comment on Lockdown Mode Configuration	49
VPN1	51
VPN2.....	51
VPN3.....	52
VPN4.....	52
Summary of Findings.....	53
Recommendations.....	54
Conclusion.....	55
Glossary.....	56

Background

In late 2018 we were involved in an incident that looked like two of our employees might have fallen victim to credential theft, using a ‘Person in the Middle’ technique referred to as a ‘pass-the-hash’ attack, also referred to in hacker parlance as a ‘Responder’¹ attack. This type of attack involves tricking a Windows host into establishing an SMB connection to a malicious host and tricking the victim into authenticating with their NTLM authentication hashes. These hashes are captured and used to crack, or in a pass-the-hash type attack.

The incident was triggered because we detected outbound SMB connections on port 445, destined for an unknown Internet host. In the end we concluded that the incident was benign. The outbound connection was caused when the workstation attempted to remap a previous connection to a corporate printer when it exited sleep mode. The corporate laptops were connected to complementary Wi-Fi at a hotel that served a DNS Search Suffix (used by the client-side resolver to “suffix” a DNS query) as part of the DHCP configuration.

The DNS Search Suffix issued by the complementary Wi-Fi served a suffix of ‘domain.com’. This resulted in the unqualified domain name of the printer (call it ‘HQ-PRINTER-1’) to resolve as ‘hq-printer-1.domain.com’. This hostname resolved to a wild-card domain called ‘domain.com’ that served the same IP address irrespective of sub-domain. Subsequently the corporate laptops attempted to map their printer to an IP address on the Internet.

We published two blog posts on the topic. The first² blog post describes the incident while the second³ blog post describes what could have gone wrong if this was an attack.

This incident occurred during a window of just seconds, during the time between the laptops returning from sleep and the corporate VPN establishing a secure tunnel that would have prevented the outbound connection ever occurring. This led us to question what protection we should reasonably be expecting VPN systems to provide for users on untrusted networks.

This paper explores the question of security for corporate endpoints on untrusted networks and examines the role that VPN technology can be expected to play, in ensuring the protection of those devices and the data they generate.

We start by discussing what it is that we expect VPNs to do for us in terms of security and consider the actual threats that a computer connected to an untrusted network might be expected to face. For the purpose of this research we have defined six distinct technical threats. These are specific attacks that would lead in one way or the other to a compromise of security and depend directly on a specific set of behaviours that can be consistently tested.

We especially examine the question of so-called ‘captive portals’, which are commonly used to control or regulate access to ‘free’ Wi-Fi services. We describe the peculiar

¹ <https://github.com/SpiderLabs/Responder>

² <https://orangecyberdefense.com/uk/blog/uncategorized/codebreak-hotel-part-one/>

³ <https://orangecyberdefense.com/uk/blog/uncategorized/dns-search-suffix-wi-fi-attacks-part-two/>

situation captive portals create for VPN technologies, where the computer is connected to the LAN or Wi-Fi (and is therefore subject to various forms of attack), but not truly connected to the Internet, in order to establish the VPN tunnel via which the computer and its traffic can be protected. Since the captive portal is essentially just a feature of the Wi-Fi access point, we argue for the purposes of this paper that any compromised (or otherwise untrusted) Wi-Fi router could essentially be made to behave like a captive portal with respect to the VPN.

We then proceed to assess how different enterprise VPN technologies deal with the peculiar situation created by captive portals. To achieve this methodically we define four different scenarios representing the diverse situations we might expect to find a corporate VPN operating under. Under each of these four sets of conditions we test for the feasibility of the six attacks we designed against computers, using four leading enterprise ‘VPN’ technologies. This results in a set of 96 different tests that collectively allow us to derive an impression of how effectively our computers are protected by VPNs when connected to untrusted or compromised Wi-Fi networks.

NOTE: At the time of publication we are continuing our work on this topic and anticipate adding more products, configurations and threat scenarios into our study. Updated findings will be published to a website we have created for this purpose as soon as we have them. Please visit our website below for our most recent findings.

<https://orangecyberdefense.com/global/insecure-remote-access/>

This paper will confirm our original concerns that the challenges created by captive portals for VPN technologies represent real technical risk. We document several scenarios in which VPNs in general cannot offer the protections we may expect, simply because the VPN cannot operate as we expect. Through systemic testing in a range of scenarios we also illustrate that several products do not offer full protection against the threat cases we identify even when they are optimally configured and working as expected. These attacks are uniquely possible because the endpoint is connected to a malicious access point that can exert some control over the configuration and behaviour of the endpoint.

Having systematically tested four major VPN products under a consistent set of conditions to examine the scope of the problem, we proceed to examine the impact of the problem by demonstrating how several contemporary attacks would be executed against systems connected from untrusted Wi-Fi networks via a VPN.

We conclude the paper by discussing how the threats and risks we identify can be mitigated, starting with the appropriate use of the features and controls offered by the VPN technologies themselves, and proceeding to an examination of the assumptions and paradigms that are commonly applied to how we safely connect our remote workers to corporate networks and systems.

At a time when more people are connecting and working remotely from free or home Wi-Fi networks than at any time before, it is essential that the primary technology we use to securely facilitate this – typically enterprise Secure Remote Access products – should offer security how and when we expect. We believe this paper will illustrate that for many

businesses this is probably not the case, and hope that this will result in a review by businesses and their vendors of the contemporary threat model for remote workers and the appropriate application of VPN technologies as a part of the response to these threats.

A note on terminology

There is a tricky issue of semantics that we need to address right up front: This paper examines the security properties of a class of technologies we will refer to primarily as ‘Virtual Private Networks’ or ‘VPN’. As our business is a significant seller and implementer of security technologies, we feel it is fair to say that this term accurately captures the class of tooling we are exploring. Indeed, it is used by some of the vendors we examine to describe themselves. The term ‘VPN’ is used much more broadly than this, however, and expands to (and even originates in) enterprise networking constructs like site-to-site links and MPLS ‘virtual private LAN’ and ‘virtual private routed network’, services amongst others. Our research and this paper do not examine the technologies deployed under this broader definition of the term ‘VPN’.

Another term used to describe the technologies we’re discussing is ‘Secure Remote Access’ or ‘Remote Access Security’. Once again, this term is widely used to describe the technologies we’re discussing here (often also by the vendors themselves) but also stretches to encompass a different class of technologies (e.g. the popular Teamviewer) that are used to remotely access desktops directly over a network. Again, this latter group is not what we are discussing here.

We will invest some page space into defining the concept of a ‘VPN’ as we understand and apply it later in this paper.

For now we content ourselves with clarifying that we are discussing ‘Virtual Private Network’ or ‘Secure Remote Access’ technologies that allow a remote user on an endpoint device like a laptop or tablet to connect into a corporate environment over an untrusted network like the Internet by enforcing authentication and creating an encrypted virtual tunnel between an agent on the computer and a gateway deployed on the perimeter of the network.

We will list some popular commercial products that match this description later in this paper.

TLDR

Executive Summary of Findings

This paper sets out to examine whether common VPN technologies, as defined above, provide the protection we expect against common threats when the endpoint is connected to an untrusted or compromised Wi-Fi Access Point.

Our findings are that out of the box and common configurations generally do not address the threats identified

The conclusions of this paper can be summarised as follows:

- We believe that the scenario where users are connecting via **compromised home Wi-Fi or malicious public Wi-Fi** is real and deserves a place on the enterprise Threat Model;
- **Captive portal** is a common scenario, but it is not an essential attribute for the threats to be real. Compromised AP or home router is just as significant;
- We believe there is a **reasonable expectation** that the ‘tunnel’ a VPN creates should **protect users** against the threats we tested;
- Out-of-the box and **common configurations** generally do not address the threats identified when the AP is considered malicious;
- All the **vendors assessed** offer features to address malicious Wi-Fi and captive portal scenario;
- However, the **effectiveness of these offerings** varied substantially and erratically across the vendors.

Standard Mode Test Outcome

Unsurprisingly, in a ‘default’ configuration state, VPNs contribute very little to protect against the threat cases we described. Furthermore, some of the VPNs tested deploy split tunnelling as the default configuration out of the box. VPNs in this configuration state remain vulnerable to most test cases even after the VPN was fully established.

(⚠ = not passed, ✅ = passed)

Test	Captured				Online			
	VPN 1	VPN 2	VPN 3	VPN 4	VPN 1	VPN 2	VPN 3	VPN 4
Network Data Leakage	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
DNS ‘person in the middle’ or spoofing	⚠	⚠	⚠	⚠	⚠	⚠	✅	✅
Harvesting credentials using spoofed website	⚠	⚠	⚠	⚠	✅	✅	✅	✅
Capturing Windows hashes via Responder	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
Using the browser as a tunnelling proxy	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
Using IPv6 to interact with host	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠

Lockdown Mode Test Outcome

The Lockdown configuration mode offered by most products to deal with the risks posted by captive portals provides a noticeable risk reduction when the VPN is established. However, the captured connection state still poses some risk and remote hosts could be vulnerable during this time, even with additional VPN configuration enabled.

(⚠ = not passed, ✅ = passed)

Test	Captured				Online			
	VPN 1	VPN 2	VPN 3	VPN 4	VPN 1	VPN 2	VPN 3	VPN 4
Network Data Leakage	⚠	⚠	⚠	⚠	✅	✅	✅	✅
DNS ‘person in the middle’ or spoofing	⚠	⚠	⚠	⚠	⚠	✅	✅	✅
Harvesting credentials using spoofed website	✅	✅	⚠	⚠	✅	✅	✅	✅
Capturing Windows hashes via Responder	⚠	⚠	✅	✅	⚠	✅	✅	✅
Using the browser as a tunnelling proxy	✅	✅	⚠	⚠	✅	✅	⚠	⚠
Using IPv6 to interact with host	✅	✅	✅	⚠	✅	✅	✅	⚠

Executive Summary of Recommendations

Our technical recommendations can be summarized as follows:

Configuration changes:

- Ensure that you understand and apply the extended configuration options provided by the VPN product with a specific view on the threats highlighted in this paper.
- Avoid using split tunnelling in your VPN configuration. Rather have corporate users' tunnel through the enterprise network where they can be subject to egress filtering, monitoring and other protections the internal network offers.
- Use your VPN configuration to enforce an internal DNS server under your control, and to hardcode the DNS Domain Search Suffix. Both the enterprise VPN products we tested offered this feature, and we expect other serious products to do so also.
- IPv6 is often overlooked as part of the threat model. It requires a complete review of existing models to understand the impact on mobile devices. The simplest approach could be to disable IPv6 until a verified solution is in place.

Other technical controls:

- Use fully qualified host names everywhere. For example, consistently use ‘ocd-src-server.ocd.local’ and not just ‘ocd-src-server’.
- Local host firewalls and sophisticated Endpoint Detection & Protection programs, properly used, can offer significant defence against the attacks described here.

Strategic thinking:

We recommend that businesses equip mobile workers with appropriate mobile data technologies and bandwidth so that they can connect via a relatively trustworthy, visible and accountable mobile network provider, rather than a veritable smorgasbord of wholly unknown free Internet providers, whose integrity and motives can never be fully trusted.

Consider Zero Trust:

Zero Trust is an emerging security paradigm in which all networks are considered equal, and untrusted, where there is no internal or external space, and where security must therefore be achieved on the endpoint and on the server without requiring a VPN.

The trade-offs we make and the reality we face.

Remote working, whether for travel, staff retention of quality of life, was a growing reality in the business world even before the COVID-19 crisis really shook things up. But with hundreds of millions of people under enforced lockdown worldwide, as Time Magazine puts it: “The cohorts working from home are about to grow into armies”⁴. This incredibly rapid and fundamental change to the working reality has predictably placed many businesses under enormous pressure to transform at a dizzying speed.



⁴ <https://time.com/5776660/coronavirus-work-from-home/>

At the heart of this enforced transformation has been the dramatic increase in demand for remote working technologies, Secure Remote Access, or so-called “Virtual Private Network” (VPN). A ‘Top 10 VPN’⁵ report on VPN demand statistics during the COVID-19 period suggests:

- Global VPN demand increased 41% over the second half of March 2020 and remains 22% higher than pre-pandemic levels.
- There are 75 countries with significant VPN demand increases since Covid-19 social restrictions began to be enacted outside China
- There are 21 countries where VPN demand surged to more than double prior levels
- The highest volume VPN demand was in the U.S. (41% peak increase), UK (35%) and France (80%)
- The largest VPN demand increases were in Egypt (224%), Slovenia (169%) and Chile (149%)
- The largest sustained increases were in Egypt (154% – over 14 days since initial peak), Peru (119% – over 28 days since peak), South Africa (105% – ongoing since mid-March).

We clearly believe that this incredible increase in demand for Secure Remote Access solutions underscores the importance of this research into the efficacy of the technology class. But it also serves to illustrate the power of the underlying systemic driver, namely that workers are increasingly connecting their computers to corporate and Internet systems via free or home Internet access points (typically Wi-Fi) and not from the office.

IT teams have scurried to address the numerous challenges that this enforced transformation has brought to light and are wrestling with how to deal with everything from remote support, attack detection and vulnerability management, on the one hand, to video conferencing (and its security), BYOD, large file transfers and dogs-in-meetings, on the other.

At the heart of all these conundrums is a fundamental question of efficiency, fairness and economy that seldom gets considered. This is the notion that home and remote workers are responsible for sourcing and paying for their own connectivity to the Internet in order to do their work. This fundamental issue does not appear to ever be explicitly vocalised, negotiated with workers, or substantially reviewed.

What is happening here in fact is that we are making a trade-off between the cost of connectivity and the cost of security. Businesses offset the cost of connecting to the office and Internet to the user, to pay for directly in their homes or seek out ‘for free’ at the various airports, hotels and coffee shops that they frequent, often with the express purpose of obtaining ‘free’ access to the Internet.

⁵ <https://www.top10vpn.com/research/investigations/covid-19-vpn-demand-statistics/>

Nothing is ever really for free, however, and where businesses avoid paying for connectivity directly, they are in fact amassing significant debt in terms of security. This paper will highlight specific examples of this kind of security debt, in the form of risks introduced by Captive portals and untrusted, malicious, or compromised ‘home’ Internet routers. This paper will highlight specific examples of this kind of security debt, in the form of risks introduced by captive portals and untrusted, malicious, or compromised ‘home’ Internet routers. As we will demonstrate in this paper, using such ‘free’ Internet access effectively requires the user to place their computer under a significant level of influence and control by the (Wi-Fi) Local Area Network and the Access Point that it creates. We can think of it as akin to asking a stranger to collect our kids from school because we do not want to pay for the gas. It is probably fine, until it is not.

Instead of addressing the obvious issues with this trade-off at their core, we accept this paradigm as a fait accompli and then seek to engineer our way out of the myriad of binds this places us in, in part by deploying technologies like ‘Secure Remote Access’ and ‘Virtual Private Networks’. Perhaps we can think of this as akin to equipping our hapless kids with a GPS watch, a flak jacket, and a can of mace to protect them in the stranger’s car. Our research presented here will seek to assess at a detailed level to what extent ‘Secure Remote Access’ or ‘VPN’ as a class of technologies actually addresses the risks presented to our corporate computers and their users.

We will argue that these technologies (especially if carelessly configured) leave the computer at significant risk, which we will recommend needs to be mitigated by introducing more stringent controls or additional security technologies like Endpoint Detection and Response (EDR), firewalls, sandboxes and the like, which will probably somewhat but not fully address the risk. This is an apparently logical sequence of deductions, but as the cost, complexity and general overhead of these security controls spiral, our (admittedly somewhat extreme) analogy hopefully illustrates that we also need to have a debate about the essence of the trade-off we’re making when users are expected to connect to the office at their own cost.

What is a VPN supposed to do?

In later parts of this document we will examine the inner workings of VPN as a technology. Perhaps more important for this paper, however, is the question of what we *expect* from a VPN technology in terms of security. Virtual Private Networks fulfil a kind of hybrid role between the networking and security domains. To a large extent the technology simply facilitates access to internal network resources. There is entrenched in this use-case, however, a set of security requirements that enable the connection to occur ‘privately’ as the name suggests.

What these security requirements are, is not always clear. Security theory would suggest that these could be reduced to the classic triad of ‘Confidentiality’, ‘Integrity’ and ‘Availability’. Certainly we expect the data transmitted between an endpoint and the target to be appropriately encrypted, resistant to tampering, and operational when required, but

this seems insufficient to describe the full set of capabilities required to allow a remote user to operate on the internal network with the same level of security as a user connected to the LAN.

As security technologies from different vendors evolve to respond to client needs, and compete with one-another, their features may no longer always map directly to the distinct set of security threats from which they emerged. This being the case, it serves us to re-examine three elements that drive security product market demand, namely the buyer's expectations, the product's features, and the authentic security threat. In a perfect market these three elements would align perfectly, but in a complex environment we believe it is safe to assume that the three elements may have diverged over time. The purpose of this research is therefore to determine in a methodical manner whether commercial VPN technologies still actually align with their clients' assumptions regarding the technical threats remote users face and the extent to which VPNs address those threats.

Later in this paper we outline a set of authentic technical threats that remote workers must contend with when connecting remotely from untrusted networks. This is an ever-shifting landscape and the specific threats we describe and test against will by no means be definitive.

The buyer's security expectations are probably less ephemeral, though perhaps a little harder to capture. For the purpose of this paper we propose that the expectation corporate users have of VPN technologies can be expressed in terms of the idea of 'equivalency':

When faced with security threats a roaming worker connected remotely via VPN would expect 'equivalent' protection to a user connected directly to the corporate LAN.

By this definition we would argue that any scenario under which a VPN-connected computer is more vulnerable to a form of attack than its LAN-connected counterpart, constitutes a security failure of the VPN.

It should be clear that a 'security failure' by this definition does not constitute an urgent exploitable vulnerability. We assert, however, that such failures in security are important to understand for security assurance. Moreover, we will demonstrate in this paper that some of these failures may in fact translate to technical vulnerabilities that can be exploited with significant downstream implications for enterprise security.

How a VPN works and what it does

Virtual Private Networks (VPN) enable endpoints to connect remotely over public networks and extend the internal network to remote locations, while ensuring confidentiality and integrity of network communications⁶. In corporate environments authentication and access

⁶ A. G. Mason, in Cisco Secure Virtual Private Network, Indianapolis, Cisco Press, 2002, p. 7.

control is an important requirement to ensure that only legitimate users gain access to corporate resources.

Commercial VPN technology has been around since the middle 1990's⁷ ⁸ ⁹. In the beginning, VPNs were mostly in the domain of enterprise businesses, government agencies, and academia. Since then there has been a steady increase in the number of commercial VPN offerings. Open source projects, such as OpenVPN, has made it possible for everyone to use a reasonably good VPN solution.

We can make safe assumptions that modern system and infrastructure design are largely based on the concept of external public networks and internal private networks segregated by a perimeter enforcement device. This assumption is based purely on the large number of vendors¹⁰ offering products that establish and manage these artificial boundaries. Firewall technologies and VPNs are prime examples of this.

Enterprise businesses equip staff with mobile devices such as laptops and smart phones to perform daily tasks. This makes the workforce much more mobile but places an implicit burden on out-of-office staff to ensure that they are always on-line. Security is handled by the underlying operating system and supporting solutions, for example a VPN. But how well does this work? Do these VPNs at least ensure confidentiality?

VPN solutions, especially enterprise grade, can be complicated and nuanced with several configuration options and combinations thereof. Things can get complicated quickly and the details depend largely on what is deployed and how it is used. Remotely supporting users with technical issues can result in overheads when trying to resolve technical issues that are caused by misconfigured solutions or troublesome software.

To put this in perspective we will need to consider in which situations VPNs are best suited to provide protection. The reality of the matter is that solutions have limitations. Sadly, that is why so many organisations end up with a plethora of layered solutions, each trying to contribute something meaningful, but only to confound the situation even further.

We would argue that the words 'Virtual Private Network' (VPN) capture exactly what the intent of the technology is, and the network part is the anchoring concept. 'Virtual' refers to the fact that the construct it creates resembles and behaves like its physical equivalent. The word 'Private' lays claim to confidentiality and implies trustworthiness. With this concept in mind we can infer that a VPN is a logical extension of a network to another geographical location, giving the illusion that a distant computing device seems to reside on the local addressable network segment. This network extension can span across public Internet.

⁷ <https://tools.ietf.org/html/draft-ietf-pppext-pptp-00>

⁸ <https://en.wikipedia.org/wiki/IPsec#History>

⁹ https://www.nrl.navy.mil/itd/sites/www.nrl.navy.mil.itd/files/files/itd_accomp_ipsec.pdf

¹⁰ <https://www.gartner.com/reviews/market/network-firewalls>

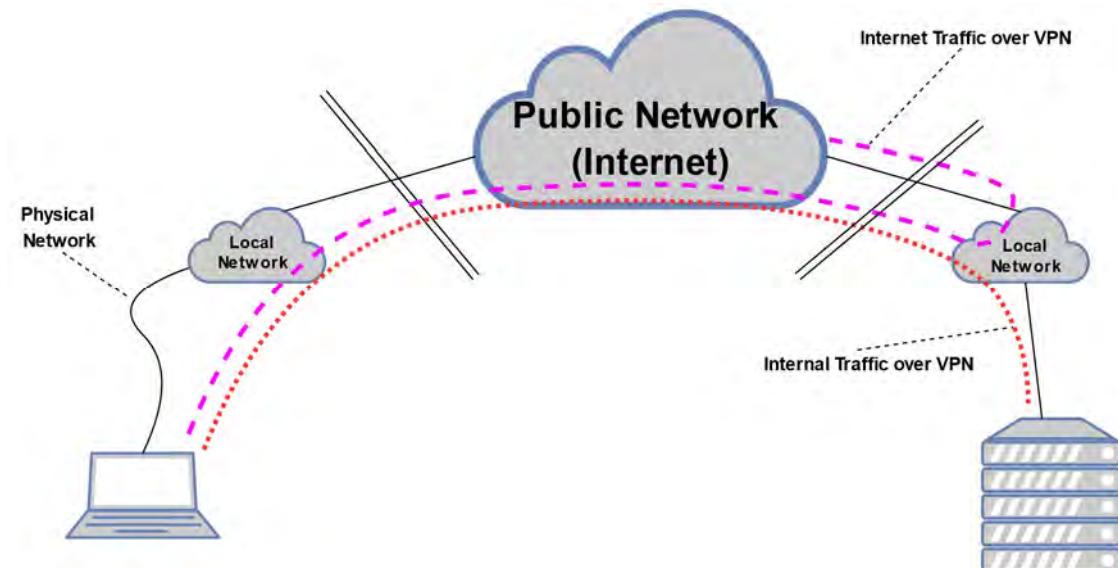


Figure 1 – All traffic tunneled over VPN¹¹

In the early days of the Internet users would use dialup networking to connect into the corporate network by utilising the plain old telephone system (POTS), using a physical medium such as a modem. Similarly, early enterprise grade leased line virtual switched digital networks such as X.25, Frame Relay, and Asynchronous Transfer Mode (ATM) networks allowed corporates to establish wide area networks linking remote branches or sites to the corporate headquarters.

Back in those days there was no channel confidentiality and channels were theoretically open to eavesdropping and interception. This effectively meant that you had to implicitly trust your telecommunications provider. In such examples data flows were restricted by logical data streams and these networks cannot be considered VPNs¹².

As the age of the Internet progressed, threats evolved, and businesses had to respond. Emphasis was placed on confidentiality, integrity, and authenticity. Out of this need the concept of the VPN was born.

VPN can be classified by ¹³:

- the **tunnelling** protocol used
- the tunnel's **termination point** location, e.g., on the customer edge or network-provider edge
- the **connection type topology**, such as site-to-site or network-to-network
- the **degrees** of security provided

¹¹ https://en.wikipedia.org/wiki/Virtual_private_network#/media/File:VPN_overview-en.svg

¹² <https://tools.ietf.org/html/rfc2132>

¹³

https://web.archive.org/web/20191115081533/https://en.wikipedia.org/wiki/Virtual_private_network

- the **OSI layer** they present to the connecting network, such as Layer 2 circuits or Layer 3 network connectivity
- the number of **concurrent** connections

Depending on the technologies used a business can have managed VPN solutions or can administer VPNs themselves. Often, depending on the definition, businesses will have some form of hybrid solution.

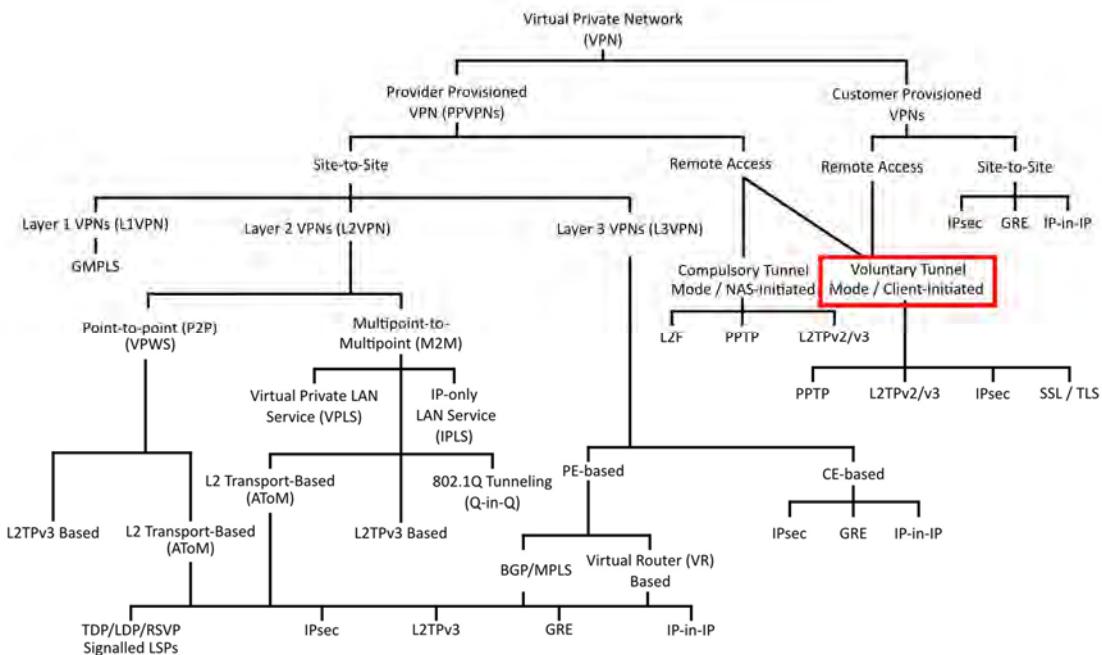


Figure 2 – Classification of VPNs¹⁴

Adding to this complication is a concept called ‘split tunnelling’. Split tunnelling is when a VPN is configured, once connected, to route specific network requests through the VPN tunnel while other traffic follows according to the default network routing rules. This is done so that only traffic destined for the corporate network is encrypted and subject to access control, while regular local network or Internet-bound traffic flows outside the VPN tunnel.

The reasoning is obvious – to allow access to resources on the local network and improve performance when accessing the public Internet. It also lessens the amount of traffic traversing the corporate network and avoids awkward issues of responsibility and accountability for traffic originating from a user connecting from home. Tunnelling all remote user traffic through the corporate VPN infrastructure can negatively impact remote users due to higher bandwidth utilisation at the central VPN service.

¹⁴ https://en.wikipedia.org/wiki/Virtual_private_network#/media/File:VPN_classification-en.svg

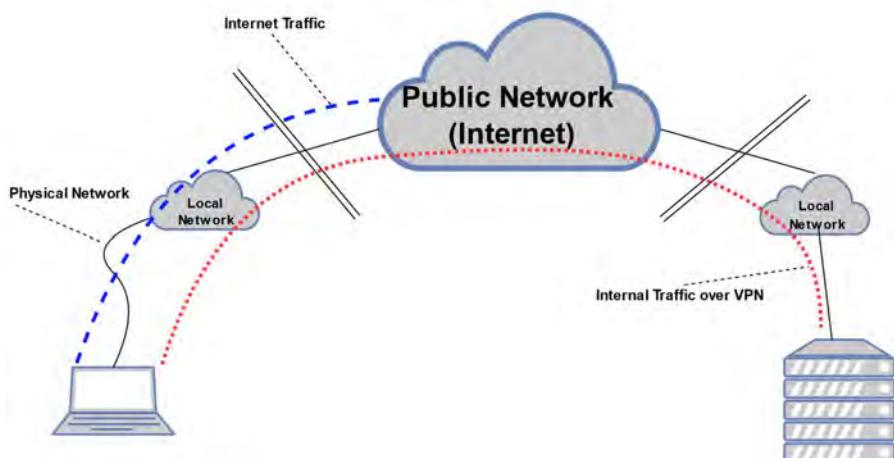


Figure 3 – Split tunnel VPN

We reiterate that for the purposes of this paper we are considering ‘Virtual Private Network’ or ‘Secure Remote Access’ technologies that allow a remote user on an endpoint device like a laptop or tablet to connect into a corporate environment over an untrusted network like the Internet by enforcing authentication and creating an encrypted virtual tunnel between an agent on the computer and a gateway deployed on the perimeter of the network.

Technical Background

Introducing captive portals

We live in a highly connected world where roaming Internet connected devices are the norm. Unfortunately, this connectivity has resource constraints and cost implications that we need to absorb or work around. Free or public Wi-Fi is most coveted by the cost conscious or those without mobile data access, typically when travelling in unfamiliar places. The necessity to be on-line and connected has become ingrained into society. Modern businesses enable staff to work remotely while on business trips or merely working from home.

Retailers, restaurants, coffee shops, the travel industry, and more, have cottoned-on to this and offer complimentary Internet access as part of their marketing and customer retention strategies. This complimentary Internet access is facilitated through Wi-Fi access points (AP) branded with the name of the respective organisation.

Some complementary Internet access requires that guests first interact with a special web server that requires either a password, voucher code, or some form of consent that involves agreeing to terms of use. The latter may even involve an exchange of user privacy for free Internet access - allowing the Wi-Fi operator to monitor network traffic and capitalise on the user's behavioural data.

We are discussing ‘Virtual Private Network’ or ‘Secure Remote Access’ technologies that allow a remote user on an endpoint device like a laptop or tablet to connect into a corporate environment over an untrusted network like the Internet by enforcing authentication and creating an encrypted virtual tunnel between an agent on the computer and a gateway deployed on the perimeter of the network.

This combination of technologies that act as a gatekeeper system and facilitates this transaction is referred to as a ‘captive portal’.

Captive portal Behaviour

On a technical level, interacting with the captive portal first requires that the guest device connects to the respective Wi-Fi AP¹⁵. The guest machine relies on the Wi-Fi AP it connected to for certain network configuration before it can interact with any network service using the Internet Protocol (IP).

The Dynamic Host Configuration Protocol (DHCP) service of the Wi-Fi AP is responsible for issuing a set of network configurations. These configurations normally consist of an IP address, subnet, gateway address, and DNS addresses that allow the guest to interact with the captive portal using a browser. DHCP can carry other information such as a connection specific DNS suffix. The guest is dependent on the DHCP response and will apply the network configuration without objection.

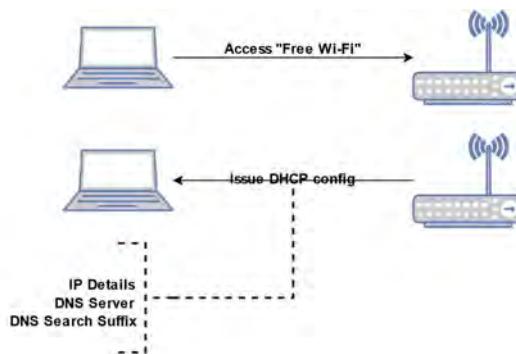


Figure 4 – Joining a captive portal Wi-Fi network

At this point the guest is free to communicate using IP, even if it is restricted to the local network only. Network traffic can route to the Internet once the conditions enforced by the captive portal has been met, if applicable.

In the case of a captive portal, a device will be able to connect to the Wi-Fi network, but it will not have Internet access. The OS of said device tests for Internet access by making an HTTP request to a predefined URL of its choosing. If the HTTP response matches what the OS expects then the OS assumes the device is connected to the Internet. If Internet

¹⁵ <https://captivebehavior.wballiance.com/>

connectivity is not yet permitted, then the response typically consist of an HTTP ‘302’ response that redirects web traffic to a web server controlled by the captive portal, where the process of registering or authorising the user is completed.

User Initiation

If the captive portal probe does not result in the expected response the OS will determine if the user can be prompted to interact with the captive portal. This phase is relevant especially to mobile devices such as phones that could be locked. Locked in this case refer to the fact that the user needs to supply biometric, pin code, password, or pattern to unlock the device to further interaction with the device.

Captive portal Presentation

The OS has already joined the Wi-Fi access point and received a DHCP address. The OS attempts to ‘check’ if it has Internet connectivity. The OS will explicitly probe to determine if there is a captive portal. It does this by opening an HTTP connection to an OS/device/browser specific URL.

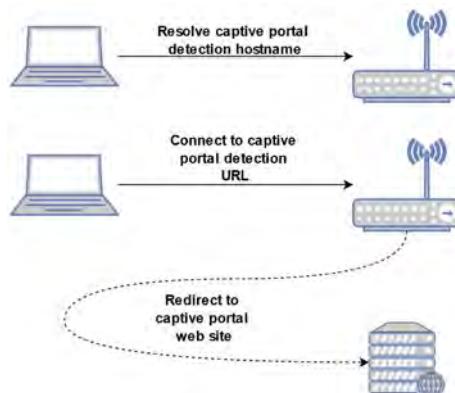


Figure 5 – Captive portal detection

The captive portal will intercept this HTTP request and issue an HTTP response. The response will typically include an HTTP 302 response code¹⁶. This will redirect to the captive portal’s web site.

The user is presented with a web browser interface that shows an HTML page from the captive portal. In the case of Android and iOS, the user is informed that a captive portal is present and asked whether the user wishes to interact with it.

Android and iOS have special web browsers built in that are called Captive Portal Mini-Browsers. These are separate from the fully-fledged web browsers typically installed for web browsing. macOS has a similar concept in the form of a Captive Network Assistant.

¹⁶ <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/302>

	Connectivity Check	User Initiation	Captive Presentation	Online State Detection
iOS	X	Device Unlock	CPMB*	X
Android	X	Push Notification -- Device Unlock	CPMB* through Push Notification	X
Windows	X	-	Default Browser	X
macOS	X	-	CNA**	X
Linux	-	-	Default Browser	-

Pre-authentication → Post-authentication
Internet connectivity state progression

*Captive Portal Mini-Browser = CPMB

**Captive Network Assistant = CNA

Windows and Linux rely on the default web browser to interact with the captive portal. Windows can automatically start the default web browser when it detects the captive portal. Linux is silent and relies on the user to start a web browser, such as Firefox, that can detect a captive portal.

The final step is for the user to fulfil the requirements of the presented captive portal page. The captive portal will lift any restrictions imposed on the guest once it is satisfied with the request.

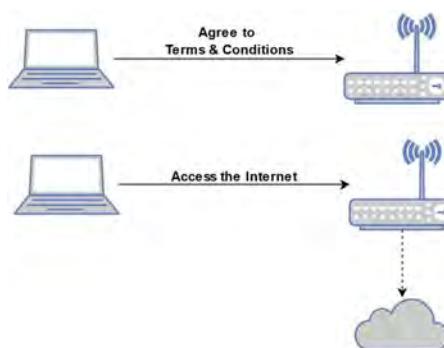


Figure 6 – Obtaining Access to Internet

Access Restrictions

A captive portal can use any number of technical capabilities to control which guest may access the Internet. This is largely up to the vendor of the captive portal to decide how

these are implemented. A very simplistic example involves a firewall and a session state controller.

The session state controller may flag the media access control (MAC) address of the guest to indicate whether it is granted Internet access. The firewall and routing rules of the captive portal will then permit traffic of the flagged MAC address to access the Internet.

The session controller will revoke access for the host when certain rules are triggered. These include idle timeout or volume of Internet data is exceeds a configured value.

Online State Detection

Completing the captive portal registration step, results in the captive portal redirecting the browser to the URL that the OS initially used to detect if it has access to the Internet.

The OS will continue to check whether it can access the Internet by requesting its test URL. Microsoft Windows 10 will for example probe <http://www.msftconnecttest.com/redirect>. These varies between vendor and product versions.

The browser handling the captive portal waits for a successful HTTP 200 response code¹⁷ to the HTTP request and checks if the request to the external website is successful. The OS will usually signal visually to the user that it could reach an Internet web site. For example, Windows will change the network icon from a globe icon to the Wi-Fi- signal strength icon.

Linux and captive portal

Our observations show that Linux does not perform captive portal detections by probing certain URLs. The end-user must know that they need to open a web browser such as Firefox or Chrome. The browser will then perform captive portal detection and will display a notification to prompt the user to interact with the captive portal¹⁸.

DNS Suffix Search List

The guest network is configured using DHCP¹⁹ and it supplies the IP addresses, subnet masks, default gateways, DNS server details and other optional values. One such option, DHCP option 15²⁰, is referred to as Domain Name under RFC 2132. Microsoft refers to it as DNS Suffix Search List or connection specific domain name suffixes²¹ and is formally defined as Dynamic Host Configuration Protocol (DHCP) Domain Search Option in RFC 3397²². This is DHCP option 119.

¹⁷ <https://developer.mozilla.org/en-US/docs/Web/HTTP>Status/200>

¹⁸ https://en.wikipedia.org/wiki/Captive_portal and <https://www.chromium.org/chromium-os/chromiumos-design-docs/network-portal-detection>

¹⁹ https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

²⁰ <https://tools.ietf.org/html/rfc2132>

²¹ [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-r2-and-2008/dd197495\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-r2-and-2008/dd197495(v=ws.10))

²² <https://tools.ietf.org/html/rfc3397>

The DNS Suffix Search List is used by Windows to lookup up hosts when an unqualified hostname is specified. The DNS client resolver will combine the host or machine name with a DNS Suffix found in the list. The DNS client will move on to the next DNS Suffix Search item until it finds a fully qualified domain name (FQDN) that resolves. This technique is used to resolve hosts when no FQDN is specified.

The DNS Suffix Search List option can be manipulated by a malicious captive portal to assist with the proposed attack scenarios listed.

Research Overview

In this section we will outline the working assumptions, overall approach and specific tests that we conducted in performing this methodology.

Working Assumptions

The work presented in this paper proceeds from several pivotal assumptions that either speak to the relevance of the work or form a basis on which our work builds. We will summarise the key assumptions here for transparency without attempting to justify the assumptions individually. The authors recognise that our eventual conclusions rest in large part on the assumptions and welcome any disagreement or debate they may spark:

- Working from home using Wi-Fi to connect to the Internet via ADSL, mobile or Fibre has become commonplace and is likely to continue growing;
- When traveling, especially abroad, mobile workers are very likely to seek out ‘free’ Wi-Fi and even more likely to willingly engage with a captive portal and make some exchange in order to do so;
- Home routers that are not expressly managed by the business should be considered insecure, or insecurely configured;
- Home routers are a target for hackers and attacks against home router technology is likely to be weaponised and performed at scale;
- There is no reason to consider any ‘free’, public or home Internet Access Point to be anything other than untrustworthy at best or malicious at worst;
- The risks inherent in connecting to insecure or malicious Wi-Fi networks are well understood and well documented;
- SSL-VPN Secure Remote Access solutions are de-facto the primary means by which these risks are addressed;
- There is an expectation (implied or inferred) that a remote endpoint connected to the corporate LAN via such a VPN should enjoy ‘security equivalency’. That is to say that they should not suffer any vulnerability of face any threat not also faced by similar endpoints directly connected to corporate LAN;
- Although they may be considered security ‘best practice’, configurations like ‘Split Tunnelling’ are considered default and are widely deployed.

Fundamental Research Question

Given the assumptions above, we summarise the fundamental question we seek to answer in this paper as follows:

If a VPN is the logical extension of a private network to another location, and if we assume that the ‘other location’ is a Wi-Fi network that is either compromised or malicious, how much protection do enterprise VPN products provide against common threats we could reasonably expect to encounter?

Research Scope

This paper will focus exclusively on customer provisioned VPNs that allow remote access by devices into corporate networks by creating a secure tunnel over the Internet.

For the purposes of this paper five common products from leading vendors were considered, namely:

- Cisco
 - Cisco ASA 9.12(3) with AnyConnect
- Pulse Secure
 - Pulse Connect Secure
 - Pulse Secure 9.1R1 Build 1505 - Server
 - Pulse Secure VPN version 9.1.1 (607) - Client
- Checkpoint
 - Check Point VPN
 - Check Point R80.30 - Server
 - Check Point VPN E81.40 Build 986101104 - Client
- Fortinet
 - Fortigate with FortiClient
 - FortiOS 6.2.4 – Server
 - FortiClient 6.4.0.1464 – Client
 - FortiClient EMS 6.2.7 – Advanced features
- Palo Alto Network
 - PAN-OS Global Protect
 - PAN-OS 9.0 (9.0.9) - Server
 - GlobalProtect 5.1.4 - Client

We consider this subset of vendors and products to be a representative sample for the class of security technologies we refer to here as ‘Virtual Private Networks’. It is with this general ‘class’ that this paper concerns itself, and not with the diverse behaviours of the individual products. For this reason, we will not present or elsewhere reveal the test results for individual products. Where we discover what we believe to be technical bugs or vulnerabilities in any given product we will engage with the vendor as appropriate, following recognised responsible disclosure protocols.

For the purpose of this experiment we limit the ‘victim’ endpoint to the Windows operating system. Our testing will examine the behaviour of these enterprise grade VPNs on Windows 10.

NOTE: At the time of publication of the first version of this report (03 August 2020) we have not completed testing for one of the five products listed above, and therefore will not include the results for that product in the results presented later in the report. This leaves the sample set for our results data at four leading products. We assert that this is a representative sample that is enough to draw conclusions from. However, we will add results for the fifth product and hopefully more when we feel confident in our findings.

Updated findings, inputs from vendors and other additional resources can be obtained by visiting our site at:

<https://orange cyberdefense.com/global/insecure-remote-access/>

The names of the vendors in the rest of this document are therefore deliberately redacted. We will refer to the four VPNs in scope using the aliases VPN1, VPN2, VPN3, and VPN4, with no relation to the order in which the products are presented in the list above.

It is important to note that most of the VPNs are components or features of a larger product set. We focused on the contributions the specific VPN components made. Any additional features were not considered.

Vendor Engagement

We have been in direct and transparent communications with all five vendors who are cooperating fully and openly with us. They are fully advised regarding the results of our research and have all been supporting our work in various ways.

All the vendors we examine in this paper have responded to our requests for input and have expressed a willingness to engage with us and even cooperate technically in the research. We note that in general the vendors have all been very interested, open and collaborative.

Test Scenarios

This section borrows from the captive portal Behaviour section. The following terminology is important as it describes the two connection states and two configuration modes a user a VPN-protected endpoint might find itself in. The combination of the two states and four modes as depicted below results in four distinct ‘test scenarios’, which we will independently test and report on in this paper.

	Standard Mode	'Lock down' mode
Captured	<ul style="list-style-type: none"> No Internet access; Most like off the shelf VPN config; Split tunnelling inactive since there's no Internet; 	<ul style="list-style-type: none"> No Internet access; Best possible working VPN config; Full tunnelling inactive since there's no Internet;
Online	<ul style="list-style-type: none"> Internet access – VPN established; Most like off the shelf VPN config; Split tunnelling enabled unless specifically discouraged; 	<ul style="list-style-type: none"> Internet access – VPN established; Best possible working VPN config; Full tunnelling;

Captured State

The capture state is symbolic of the fact that guests are trapped in a walled garden. Guests have very limited network interaction and there is no Internet access. This is the default connection state when guests join a captive portal network and could be emulated by an attacker who controls any Wi-Fi access point.

The guest must typically interact with a web page that requests consent, accept terms and conditions, provide payment, or verify identity. Only when the captive portal is satisfied with the supplied information will the guest be permitted to move to the next connection state. An attacker could achieve the control offered by a captive portal on a compromised Access Point without making this apparent to the user.

Online State

This state signifies that the guest may route traffic to services beyond the captive portal. In other words, the guest has Internet access.

The VPN can only establish once the guest is in the Online State.

VPN in Standard Mode

Standard Mode is a term that we chose to describe a probable ‘default’ configuration state of the VPN. This state normally requires the least possible number of steps to get a connection between the remote host and the VPN gateway.

We based this state on what we (at our own discretion, in consultation with our various product specialists) consider to be the most likely, minimum configuration a business would be required to support remote workers.

In ‘standard mode’ we will follow any reasonable guidance that vendor documentation imparts regarding suggested setup of its VPN product.

If vendor documentation states that certain configuration is preferable, then we will make changes to follow that guidance even if it is not out of the box configuration. For such guidance to be considered the vendor documentation must be available on the vendor’s website and should be available to anyone without having to register or have a support account.

VPN in Lockdown Mode

Modern VPN technologies have responded to the challenge of captive portals as described earlier by introducing a set of features generally known as ‘captive portal remediation’, ‘lockdown mode’ or ‘hotel mode’, a collection of features which are supposed to provide better protection in walled garden environments. Products we examined that did not explicitly label these features nevertheless still provided a set of controls across the configuration option set that would result in an equivalent of ‘lockdown mode’ being achieved.

‘Lockdown Mode’ can be thought of as a set of VPN features that are designed to limit the amount of traffic that leaves the endpoint while it is on the wireless LAN, dealing with the captive portal.

Some of these features are also designed to limit the interaction that the host with the VPN client in lockdown mode can entertain.

Some vendors do not refer to the term ‘lockdown’ or ‘captive portal remediation’ explicitly. It is still possible to enable a subset of features that speak directly to the role the VPN plays in mitigating the test cases we present.

Lockdown Mode should not be confused with best practices. Best practices speak to wider and richer configuration set that includes other components not specifically covered by this paper.

Target Security Model

Our model assumes that any Wi-Fi Access Point not under the control of the business is untrusted therefore needs to be treated as malicious. The influence and control that a malicious Access Point exerts over the endpoint is perfectly illustrated by the captive portal. We therefore use the scenario of a malicious captive portal to illustrate the threats we illuminate in this report. The captive portal environment forces guests into a compromising position that benefits an attacker. All traffic transmitted when connected to the captive portal may be intercepted, modified and relayed. The captive portal may also present the guest with web pages that aim to phish the user or otherwise manipulate or compromise the browser.

VPNs are expected to guarantee confidentiality and integrity of tunneled data when in transit. In addition, VPN solutions typically enforce access control on the perimeter. This combination of features has made “VPNs” the standard form of Secure Remote Access technology provided to remote users when connecting over untrusted networks.

The guest, upon joining the Wi-Fi network of the captive portal, is issued DHCP configuration. These parameters contain a wide array of options that are under the control of the attacker.

It is apparent, given the architecture and assumptions described above, that when a guest is in the process negotiating with a captive portal, that there exists a period where communications between the guest and the captive portal does not pass through the VPN.

With these assumptions in mind, the threat model should be able to answer the following questions in terms of:

- **Confidentiality**
 - How much unsolicited network traffic is broadcast by the guest while associated with the local network of the AP?
 - What role does dynamic network configuration fields, such as connection specific DNS suffixes, play in leaking network traffic.
 - How much network traffic is leaked to the local network of the AP while connected to the VPN?
 - Does the guest leak any information that is considered sensitive or critical?
- **Integrity**
 - Can the client applications on roaming devices detect person-in-the-middle attacks?
 - How resilient are roaming devices against credential theft?
- **Access Control**
 - Can attackers use guests on the malicious free Wi-Fi to tunnel over the VPN into the corporate network?

Technical Threat Scenarios

Under these circumstances described above, considered in light of the working assumptions described earlier in this document, we define six technical threats that form the basis of our testing. These technical threats are as follows:

 <h3>Sniffing sensitive data</h3> <p>This threat involves the attacker extracting sensitive information like login credentials from endpoint network traffic. For the purposes of our research we consider any data unique to business, endpoint or user, to be 'sensitive'. This definition should make it clear that we do not undertake to prove that this information is exploitable by the attacker.</p> <p><i>We consider it sufficient that such data can be collected by the attacker to demonstrate a failure of security.</i></p>	 <h3>DNS ‘person in the middle’ (PiTM) or spoofing</h3> <p>The attacker feeds fake DNS responses to legitimate requests from the client, thereby controlling where the subsequent connection ultimately terminates. This is a precursor to several other attacks, like spoofed websites.</p> <p><i>We consider any instance where we as the ‘attacker’ can define the IP address returned for a legitimate DNS query to demonstrate a failure of security.</i></p>
 <h3>Harvesting credentials using spoofed website</h3> <p>Once the attacker controls DNS and routing (as they would with a malicious AP) they can present the user with a fake login page to valuable resources like O365 to harvest login credentials. The important thing to note here is that for the purposes of this research such a ‘phishing’ attack should be uniquely possible because the victim is connected to a malicious Wi-Fi LAN.</p> <p><i>We consider any instance where the attacker can present the user with a fake login page for a legitimate site before she can benefit from any protections offered by the corporate network to be failure of security.</i></p>	 <h3>Capturing Windows hashes via Responder</h3> <p>‘Responder’ attacks involve tricking Windows systems into connecting to a fake Windows service, which in turn requests authentication and then captures the password hash that is sent. This enables further attacks against Active Directory. It should be apparent that obtaining such a hash would still require an attacker to either crack or pass the hash in a second phase of attack and have access to an appropriate interface via which she can do so. We do not include this second in our testing.</p> <p><i>We consider it sufficient that an endpoint be persuaded to present the user’s hashed password to a system controlled by the attacker to demonstrate a security failure.</i></p>



Using the Browser as a tunnelling proxy

Once the attacker controls DNS and routing (with a malicious Access Point) they can inject code like JavaScript into legitimate websites to exert remote control over the victim's computer, for example to abuse it as a pivot point to tunnel traffic into the corporate network. It is apparent that such JavaScript injection is complex, provides the attacker with limited control and is strongly countered by browser security models whenever possible.

We consider the ability to inject JavaScript code into user's browser by virtue of their location on the malicious Access Point or interaction with the captive portal sufficient to demonstrate a security failure.



Using IPv6 to interact with host

Most enterprise VPN technologies are designed to protect IPv4 traffic, but many endpoints now also run IPv6 stacks that can be used to communicate on the LAN and Internet. If the VPN doesn't control IPv6, that presents the attacker with an open channel for communicating with the computer. It is apparent that simple interaction with the IPv6 stack of an endpoint does not imply that a compromise of that host follows.

For the purpose of this research we consider any ability to interact with the IPv6 stack of an endpoint when similar interaction with the IPv4 stack is not possible, to be a security failure.

Research Methodology

A Windows 10 host, representing the victim, will connect to the malicious Wi-Fi network that act a captive portal. For each VPN, the targeted user will attempt to start the VPN with configuration as per our Standard Mode definition. The targeted user will be subjected to defined test cases that will determine the behaviour of the respective VPNs in that mode.

Next, we will apply the configuration that places the tested VPNs in Lockdown Mode. The targeted user will, for each VPN, attempt to establish a VPN session. As in the Standard Mode test phase, the target will be subjected to the same attacks.

The six tests defined above (and detailed below) are repeated for Capture State and Online State. This brings us to four test scenarios with six possible tests for each scenario across four VPN products. Test case 5 spans across the Capture and Online state as it requires the VPN to be established.

The table below illustrates how the five products can be tested against the six threat scenarios defined in each of the 4 test scenarios to produce a total of 96 results.

		Standard Mode						'Lock down' mode						
		T-1	T-2	T-3	T-4	T-5	T-6		T-1	T-2	T-3	T-4	T-5	T-6
Captured	VPN 1	1	2	3	4	5	6		25	26	27	28	29	30
	VPN 2	7	8	9	10	11	12		31	32	33	34	35	36
	VPN 3	13	14	15	16	17	18		37	38	39	40	41	42
	VPN 4	19	20	21	22	23	24		43	44	45	46	47	48
Online	VPN 1	49	50	51	52	53	54		73	74	75	76	77	78

VPN 2	55	56	57	58	59	60		79	80	81	82	83	84
VPN 3	61	62	63	64	65	66		85	86	87	88	89	90
VPN 4	67	68	69	70	71	72		91	92	93	94	95	96

Test Environment

Windows Client Environment

The OS was Windows 10 Version 1903 build 18362.535 and ran as a virtual machine on VMWare Workstation 14.3.

The network adapter used to communicate with the captive portal was a TP-Link TL-WN722N USB Wi-Fi dongle.

Windows Host Configuration

Wi-Fi Network Profile

The Wi-Fi Network profile on the Windows host should be set to Public. For example:



Ensure the Public network is selected:

Network profile

Public

Your PC is hidden from other devices on the network and can't be used for printer and file sharing.

Private

For a network you trust, such as at home or work. Your PC is discoverable and can be used for printer and file sharing if you set it up.

[Configure firewall and security settings](#)

Windows Firewall

The firewall associated with the Public network profile must be turned off. The purpose of this is to determine if the VPN client adds any additional protection on top of the firewall. It is also needed to test any leakage when the full tunnel is enabled.

To modify the firewall settings. On the network profile screen click “Configure firewall and security settings”.

Network profile

Public

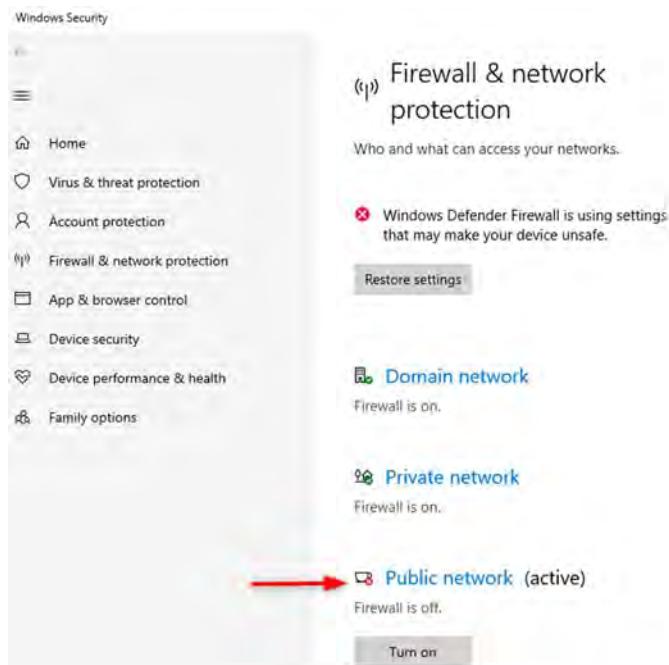
Your PC is hidden from other devices on the network and can't be used for printer and file sharing.

Private

For a network you trust, such as at home or work. Your PC is discoverable and can be used for printer and file sharing if you set it up.

[Configure firewall and security settings](#)

You will be asked to switch applications. Accept the request. The following will be displayed:



Click Public network and you will be displayed with something resembling the following:



Make sure the Public firewall is off and “Block all incoming connections” is unticked.

Windows Virus and Threat Protection

Real-time protection can be enabled. The rest should be disabled.

Virus & threat protection settings

View and update Virus & threat protection settings for Windows Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

Cloud-delivered protection

Provides increased and faster protection with access to the latest protection data in the cloud. Works best with Automatic sample submission turned on.

Automatic sample submission

Send sample files to Microsoft to help protect you and others from potential threats. We'll prompt you if the file we need is likely to contain personal information.

IPv6 Stack

In our testing environment we have assumed that IPv6 is installed and enabled on the victim endpoint. We consider this to be a ‘common’ configuration, but also reasonable and necessary for the test cases involving IPv6 described earlier.

Captive portal

In this scenario the captive portal is under the control of the attacker.

The captive portal is a Mikrotik RouterBoard 751G-2HnD with RouterOS version 6.43.16. The DHCP settings are served by the Mikrotik captive portal. This means that the attacker can control the DNS server IPs and connection specific DNS suffix.

In the IPv6 test case (Test 6) the DHCP settings are served by the Kali Linux host.

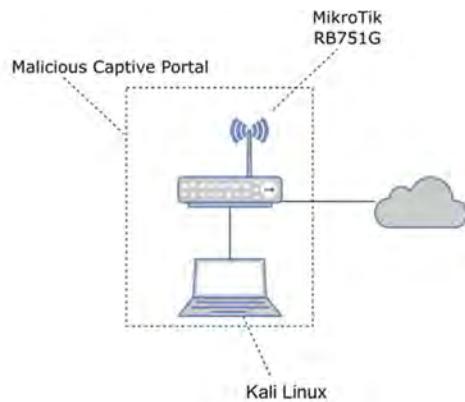


Figure 7- Malicious captive portal

The captive portal router has one Wi-Fi adapter and 5 ethernet ports. The Wi-Fi adapter acts as an access point that guests use to access the captive portal. One ethernet port on the captive portal acted as the default gateway and connected to another device that provided Internet access. A different ethernet port on the captive portal was setup to act as a port mirror. It forwarded all Wi-Fi traffic seen by the access portal to a passive collector under control of the attacker. The remaining three ethernet ports were not connected to anything.

Attacker Platform

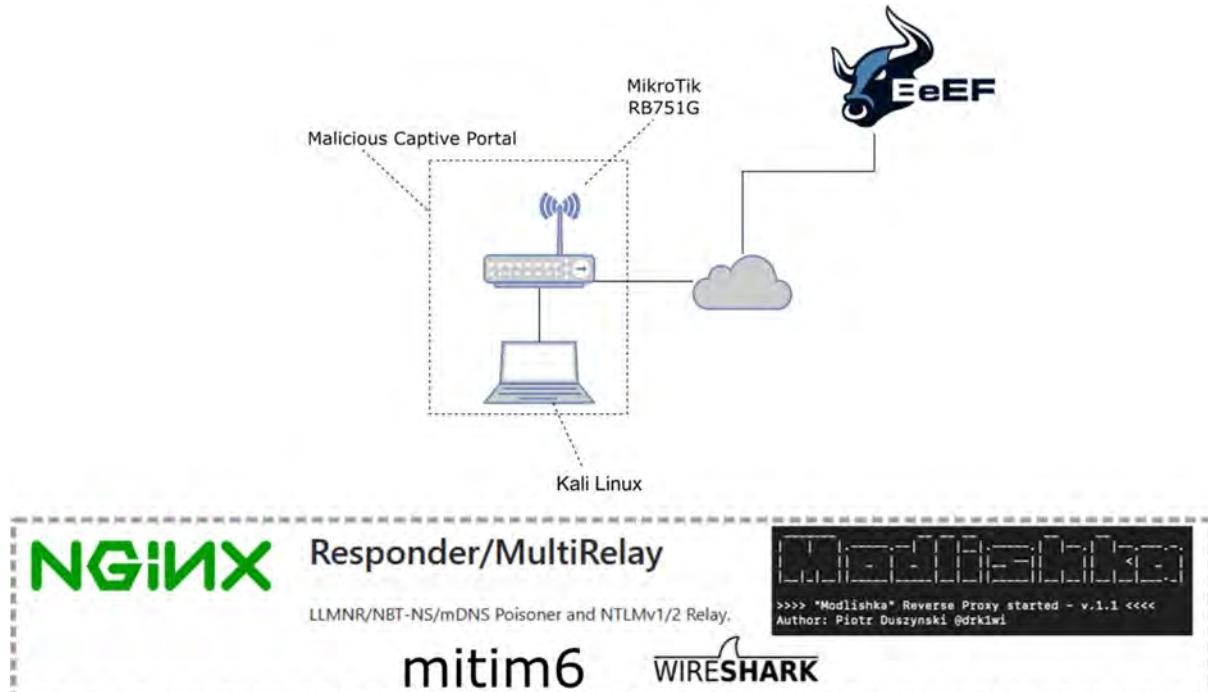
Complementing the captive portal is a laptop with Kali Linux connected. This laptop is connected to the captive portal using two network interfaces, namely Wi-Fi and ethernet.

The Wi-Fi adapter is used to interact with any clients of the captive portal’s network once these have passed the initial captive portal authentication phase. The Wi-Fi connection

from the Kali box is used to launch active attacks against the guests of the captive portal. Responder was the main attacker tool used to steal Windows authentication hashes.

The ethernet adapter is connected to the ethernet port of the Mikrotik router. The ethernet port on the Mikrotik router is configured to duplicate or mirror the Wi-Fi network traffic. Wireshark, a network traffic capturing tool, will record the network traffic seen by the Mikrotik Access Point, specifically the Wi-Fi.

An external host, with direct Internet access, was used to host the Browser Exploitation Framework (BeEF) server. This did not run on the Kali Linux host.



Software Used

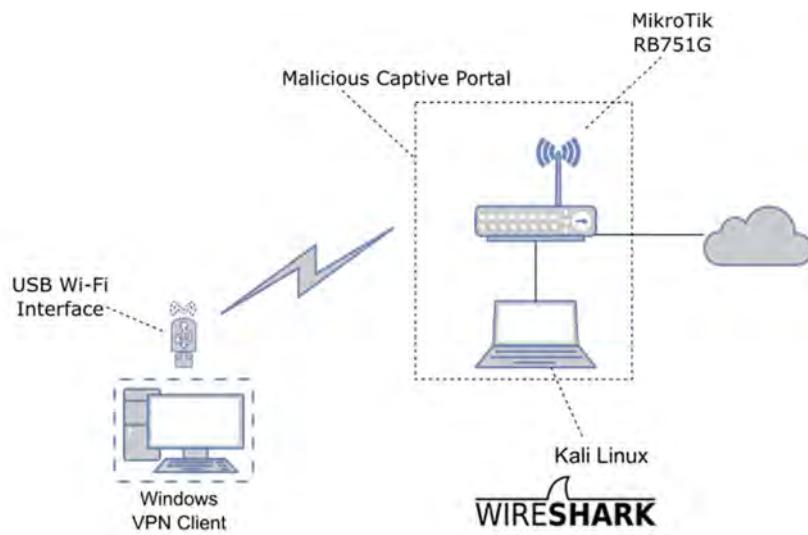
- Responder - <https://github.com/lgandx/Responder>
- Modlishka - <https://github.com/drk1wi/Modlishka>
- BeEF - <https://github.com/beefproject/beef>
- Mitm6 - <https://github.com/fox-it/mitm6>
- Wireshark - <https://wireshark.org>
- Nginx -<https://nginx.com>

Test Cases

We will present six test cases that we executed to review the behaviour of the respective VPN products.

Test Case 1 – Sniffing Data

This test case seeks to identify any apparently sensitive data transmissions originating from the Windows host connected to the captive portal. For the purposes of this test we define ‘sensitive’ as any information unique to business, user or endpoint, that would not be seen if the endpoint was connected directly to the corporate LAN. This could include domain names, SMB share mappings, or information that could be used assist targeting in some way.



This test is applicable to all VPN configuration modes in either captured or online state.

Test Passed:

There are no packets originating from the Windows host in the Wireshark capture running on the Kali Linux host.

Test Failed:

Inspect the Linux Wireshark capture and identify if it contains any packets originating from the Windows host.

Clear Fail: NETBIOS packets originating from Windows host.

Acceptable: DHCP and DNS does not constitute a failure condition.

Step 1:

Start Wireshark – select the applicable interface on:

- Kali Linux host
- Windows Host

Step 2:

Connect Windows Host Wi-Fi to the captive portal SSID.

Wait +- 1 minute.

Expected: A web browser should automatically appear with the captive portal interface.

Step 3:

After the web browser appears, showing the captive portal interface, let both Wireshark captures run for +-1 minute.

Step 4:

Pass the captive portal to get Internet access.

Step 5:

Verify that the VPN connect else force the VPN to connect. Note the time the VPN establish.

Step 6:

Wait +- 1 minute after the VPN established to let Wireshark capture packets.

Step 7:

Review the Test Passed or Test Failed conditions.

Test Case 2 – DNS ‘Person-in-the-Middle’ / Poisoning

The host connecting to the captive portal must use the DHCP and DNS services provided if the host wishes to satisfy the captive portal. This puts the captive portal in a position that enables it to steer hosts away from legitimate services.

This test is applicable to all VPN configuration modes in either captured or online state.

Test Passed:

- Captive State
 - It is unlikely that this test can be passed as the DNS server can serve any data to the client, which must be accepted.
- Online State:
 - The DNS cache of the host connected to the web service should be ideally disabled or the cache must be flushed before the VPN is established.

Test Failed:

The DNS cache of the Windows client holds resolved hostnames and IPs. Specifically, we are looking for fully qualified domain names with the connection specific DNS suffix supplied by the DHCP service of the captive portal. These will have very high TTL, for example 8600 plus seconds.

These are assigned by the DNS server under control of the malicious captive portal.

Step 1:

Start Wireshark on the Windows host and select the WLAN interface.

Step 2:

Open a command prompt on Windows Host and run:

ipconfig /displaydns

Nothing should be returned. Keep this command prompt open.

If entries were returned, then run:

ipconfig /flushdns

Now run:

ipconfig /displaydns

Nothing should be returned. This test cannot continue if DNS entries populate the result set.

Keep this command prompt open.

Step 3:

Connect Windows Host's Wi-Fi to the captive portal SSID.

Wait +- 1 minute. Follow any instruction by the VPN client to ensure the captive portal opens in a web browser.

Step 4:

Enable or prepare the wild card DNS resolver to resolve any fully qualified hostnames that matches the DNS search suffix. This is where the configured DHCP domain name (option 15) or DNS search list (DHCP option 119) comes into play.

The wild card domain should resolve with the IP of the Kali Linux host.

Step 5:

Run the following command in the command prompt:

ping blackhat

This unqualified hostname should resolve by using the DHCP assigned search suffix and with a wild card hostname DNS resolver on the malicious captive portal.

Step 6:

Pass the captive portal by submitting any required information and gain Internet access.

Step 7:

Wait for the VPN to establish. If it does not automatically connect, then force it online.

Step 8:

Run the following command in the command prompt:

ping blackhat

This should not resolve, and the ping is expected to fail.

Step 9:

In the existing open command line window run:

ipconfig /displaydns

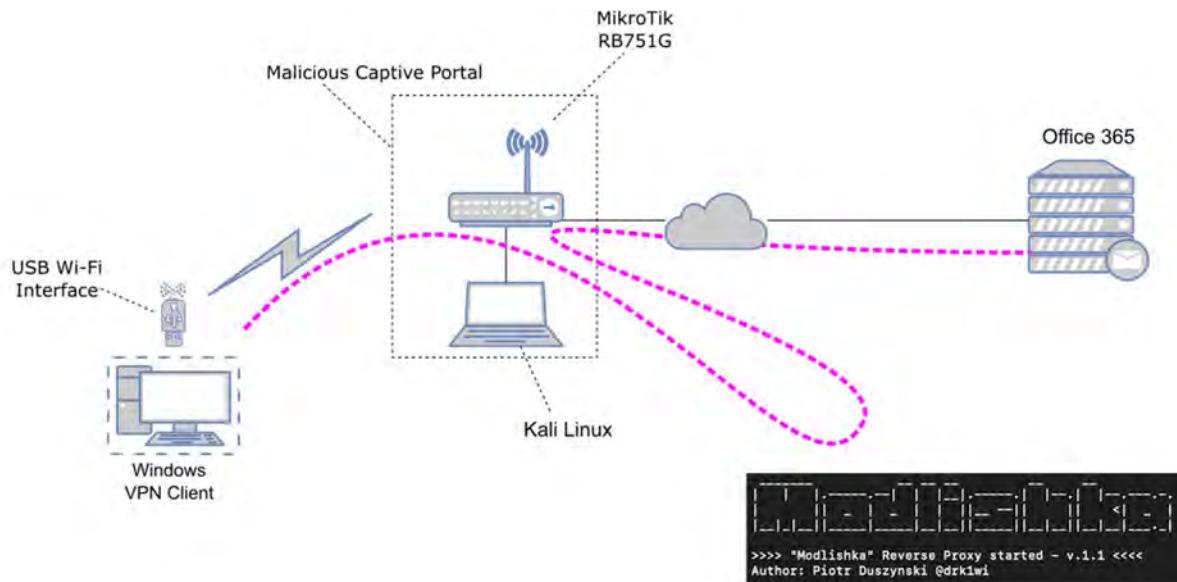
Step 10:

Review the Test Passed or Test Failed conditions.

Test Case 3 – Credential Harvesting

This test case attempts to steal credentials and perhaps even a session token of an Office365 account. The test, in its current form, relies on the ability to initiate an Office365 login session using a plain HTTP request to office.com.

A tool called Modlishka is used to perform a ‘person-in-the-middle’ attack.



This test is applicable to all VPN configuration modes in either captured or online state.

Test Passed:

The office.com web site cannot be interfered with or there are no credentials capture on the Modlishka dashboard.

Test Failed:

The Modlishka dashboard shows a username, password, and session token that can be replayed.

Specific Requirements:

- Modlishka
 - Modlishka configuration is required
- Let's Encrypt Wildcard certificate is required.
- DNS Config that resolve office.com hostname with the IP of the Kali Linux host running Modlishka.

Additional Test Case Execution Notes:

This test case must be repeated twice. Once for Captured state and once for Online state. Pay attention to Step 2.

Clear the browser cache that will be used to launch the office.com web site for each connection state test.

Step 1:

On the Kali Linux host open a bash shell and run Modlishka.

Step 2:

On the Windows host connect to the captive portal SSID.

- Captive State:
 - Skip this and proceed to the next step.
- Online State:
 - Wait +- 1 minute. Follow any instruction by the VPN client to ensure the captive portal opens in a web browser.
 - Supply any information to get past the captive portal so that it grants access to the Internet.

Step 3:

Activate the DNS configuration to map office.com to the IP of Kali Linux host running Modlishka.

Activate any configuration that will allow FQDN hostname ending with a search suffix we configured to resolve with the IP of the Kali Linux host.

Step 4:

Open a web browser, if one is not yet open, and in the address bar and enter <http://office.com>.

The web browser should now show the Microsoft office.com web site. If DNS poisoning helped and it directed the browser to the Kali Linux host then the address bar will show the rewritten address supplied by Modlishka.

Step 5:

Continue with the Office365 login process.

Step 6:

Next, we need to verify that Modlishka captured the credentials and the session token.

Use a web browser to access the Modlishka dashboard.

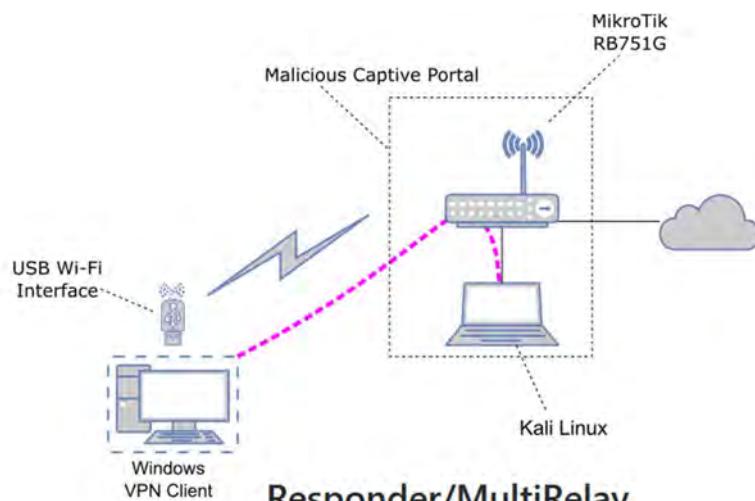
Step 7:

Review the Test Passed or Test Failed conditions.

Test Case 4 – Windows NTLM Hashes Theft with Responder

This test case aims to verify if the Windows host will exchange a NTLM hash with the Responder running on the Kali Linux host.

The goal is to determine if the VPN plays any role in preventing any interaction with any services on another host.



LLMNR/NBT-NS/mDNS Poisoner and NTLMv1/2 Relay.

This test is applicable to all VPN configuration modes in either captured or online state.

Test Passed:

The Responder service on the Kali Linux host must not have any NTLM hashes from the Windows host.

Test Failed:

If the Responder application shows a NTLM hash that originate from the Windows host.

Specific Requirements:

- Responder

Step 1:

On the Kali Linux host open a bash shell and run:

```
./Responder.py -I wlan0
```

Step 2:

Start Wireshark on and select the applicable network interface:

- Windows host
- Kali Linux

Step 3:

On the Windows host connect to captive portal SSID.

Follow any instruction by the VPN client to ensure the captive portal opens in a web browser.

Step 4:

Open a command line window and run the following command:

```
net use \\blackhat\\testtest
```

If a prompt is shown go check if there is hash printed in the Responder window on the Kali Linux host.

Step 5:

Pass the captive portal to get Internet access.

Step 6:

Verify that the VPN establishes a connection, else force the VPN to connect.

Step 7:

Wait +- 1 minute after the VPN established to let Wireshark capture packets.

Step 8:

Open a command line window and run the following command:

```
net use \\blackhat\testtest
```

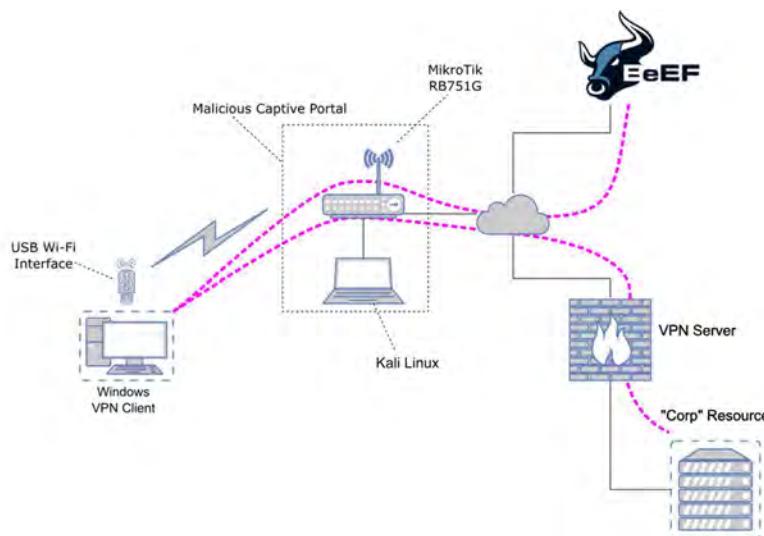
If a prompt is shown go check if a hash was printed in the Responder window on the Kali Linux host.

Step 9:

Review the Test Passed or Test Failed conditions.

Test Case 5 – Browser Tunnelling with BeEF Hooks

This test case uses the Browser Exploitation Framework called BeEF. We use the Modlishka tool, see Test Case 3, to inject JavaScript into the Office365 page. This JavaScript is the BeEF Hook and it will connect back to our command & control server.



Note: This test is only applicable when the VPN established a tunnel and can only be tested once Internet access was granted.

Test Passed:

Online State: The BeEF Hook JavaScript could not connect back to the command and control server or the Beef session is lost once the VPN is established.

Test Failed:

Online State: The BeEF Hook JavaScript connects to the command and control server. We can use the WebRTC feature to 'scan' a host on the simulated corporate LAN that is reachable when the VPN is up.

Specific Requirements:

- BeEF Server
 - Capability to host BeEF command and control server with Internet facing IP.
- Modlishka
 - Modlishka configuration is required.
- Let's Encrypt Wildcard certificate is required.
- DNS Config that resolve office.com hostname with the IP of the Kali Linux host running Modlishka.
- DNS Config that resolve any FQDN hostname that matches our domain name (DHCP option 15) or DNS search list (DHCP option 119) with the IP of the Kali Linux host running Modlishka.

Step 1:

Start the BeEF server on a hosting service with Internet facing IP.

Step 2:

Start the Modlishka server that is configured to inject the BeEF Hook. Ensure that Nginx is also running. Verify that it hosts the interim BeEF Hook that Modlishka will inject a reference to.

Step 3:

On the Windows host connect to captive portal SSID.

Step 4:

Open a web browser, if one is not yet open, and in the address bar type *http://office.com*.

The web browser should now show the Microsoft office.com web site, but the address bar will show the rewritten address supplied by Modlishka.

Step 5:

Start a web browser if it has not already opened with the captive portal prompt.

Supply any information to get past the captive portal so that it grants access to the Internet.

Step 6:

Wait for the VPN to establish.

Step 7:

Open the BeEF console and identify if the 'hooked' web browser has connected to the C2 server.

Step 8:

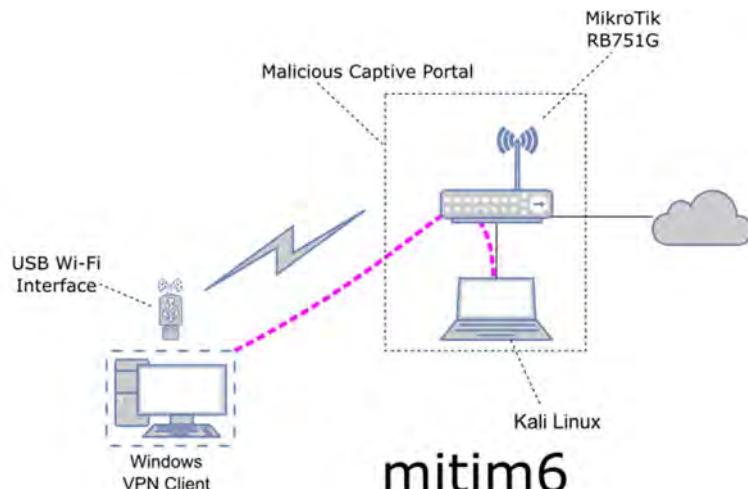
Using the BeEF command and control interface issue a command to scan an internal IP on the ‘corporate’ network.

Step 9:

Review the Test Passed or Test Failed conditions.

Test Case 6 – Interact with host using IPv6

This test case aims to verify if the Windows host can at some rudimentary level be interacted with using IPv6. The Kali Linux host will play the role of DHCP server and issue DHCPv6 addresses to any party on the network that broadcasts a DHCP discovery packet.



Now that we have issued the host with an IPv6 address we test if we can use IPv6 to ping the host.

This test is applicable to all configuration modes in either captured or online state.

Test Passed:

The Linux host cannot ping the IPv6 address it severed the Windows host.

Test Failed:

The Linux host can ping the IPv6 address it severed the Windows host.

Specific Requirements:

- Mitm6 - <https://github.com/fox-it/mitm6>

Step 1:

On the Kali Linux host start the mitm6 service.

Step 2:

Start the Wireshark application on:

- Windows host
- Kali Linux

Step 3:

On the Windows host connect to the captive portal SSID.

Step 4:

In the Kali Linux host running mitm6 – note the IPv6 address issued to the Windows host.

Step 5:

Open a command prompt and run:

ipconfig /all

Note the IPv6 address assigned to the Wi-Fi adapter. Compare if the IPv6 values match for that adapter.

Step 6:

Open a bash shell on the Kali Linux and run:

ping6 <IPv6 address of Windows Host Here>

The presence of an ICMP6 response will determine the outcome of the test.

Step 7:

Start a web browser if it has not already opened with the captive portal prompt.

Supply any information to get past the captive portal so that it grants access to the Internet.

Step 8:

Wait for the VPN to establish.

Step 9:

Open a command prompt and run:

ipconfig /all

Note the IPv6 address assigned to the Wi-Fi adapter. Compare if the IPv6 values match for that adapter.

Step 10:

Open a bash shell on the Kali Linux and run:

`ping6 <IPv6 address of Windows Host Here>`

The presence of an ICMP6 response will determine the outcome of the test.

Step 11:

Review the Test Passed or Test Failed conditions.

Research Findings

The following sections provide a summary of observations of features and behaviours of tested products in the various states and modes.

Standard Mode Test Outcome

Unsurprisingly, in this configuration state VPNs contribute very little to protect against our threat cases as described above. Furthermore, some of the VPNs tested deploy split tunnelling as the default configuration out of the box. VPNs in this configuration state remain vulnerable to most test cases even after the VPN was fully established.

(⚠ = not passed, ✅ = passed)

Test	Captured				Online			
	VPN 1	VPN 2	VPN 3	VPN 4	VPN 1	VPN 2	VPN 3	VPN 4
Network Data Leakage	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
DNS ‘person in the middle’ or spoofing	⚠	⚠	⚠	⚠	⚠	⚠	✓	✓
Harvesting credentials using spoofed website	⚠	⚠	⚠	⚠	✓	✓	✓	✓
Capturing Windows hashes via Responder	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
Using the browser as a tunnelling proxy	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠
Using IPv6 to interact with host	⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠

Comment on Standard Mode Configuration

The standard mode configuration offers limited guarantees of confidentiality. Only traffic that specifically traverses the VPN can be considered confidential.

All VPNs we evaluated allowed some form of interaction with the LAN associated with the captive portal after the VPN connected. This leaves the remote host open to further attack.

These DNS requests, including multicast DNS requests, may reveal resource names that the remote host is attempting to contact over the VPN. This allows an attacker to harvest information about the guest as well as inviting it to connect to services under the control of the attacker based.

In standard mode, the VPN must be connected to make any noticeable contribution to the overall risk reduction effort.

Lockdown Mode Test Outcome

The Lockdown configuration mode provides a noticeable risk reduction when the VPN is established. The Captured connection state still poses some risk and remote hosts could be vulnerable during this time, even with additional VPN configuration enabled.

(⚠ = not passed, ✅ = passed)

Test	Captured				Online			
	VPN 1	VPN 2	VPN 3	VPN 4	VPN 1	VPN 2	VPN 3	VPN 4
Network Data Leakage	⚠	⚠	⚠	⚠	✅	✅	✅	✅
DNS ‘person in the middle’ or spoofing	⚠	⚠	⚠	⚠	⚠	✅	✅	✅
Harvesting credentials using spoofed website	✅	✅	⚠	⚠	✅	✅	✅	✅
Capturing Windows hashes via Responder	⚠	⚠	✅	✅	⚠	✅	✅	✅
Using the browser as a tunnelling proxy	✅	✅	⚠	⚠	✅	✅	⚠	⚠
Using IPv6 to interact with host	✅	✅	✅	⚠	✅	✅	✅	⚠

Comment on Lockdown Mode Configuration

Test 5, which involves using the browser to tunnel through the VPN, leverages the captured state to highlight the possibility of threats bridging over into the VPN tunnel. It is possible to take advantage of the opportunity presented in the captured state to meddle with content served to the browser *before* the VPN is established. This someone theoretical threat is demonstrated by our research.

The Captive Portal Mini Browser is a distinctive feature of Lockdown Mode. This is a common feature of mobile operating systems such as Android and Apple’s iOS. Some VPNs have taken to this feature to help limit the impact the malicious captive portal can have on the session.

The lockdown configuration for VPN1 and VPN2 included the Captive Portal Mini Browsers to interact with the captive portal. VPN1, however, still allowed the user to interact with web resources on the LAN using a different web browser. As we are looking at the configuration

options that speak directly to our test cases we consider this a ‘pass’, however further examination of the implications of this ‘feature’ may warrant further investigation.

VPN2 was strict enough to limit interaction only through its Captive Portal Mini Browser. This offers appropriate protection against the tests where we attempted to harvest credentials, and off the back of that, tunnel through the VPN.

The Captive Portal Mini Browser presented by VPN1 seems to be proprietary. The Captive Portal Mini Browser of VPN2 seems to embed something resembling Internet Explorer or a neutered variation thereof. We did not explore the limitation of these Captive Portal Mini Browser, but it seems like further investigation may be warranted.

VPN3 and VPN4 did not have an option for Captive Portal Mini Browsers. They relied on the default web browser configured. This presents an opportunity for other types of attacks that could poison the browser cache or leverage weakness in the browser.

Here is a list of configuration options we considered important for Lockdown Mode to be effective:

- Use Captive Portal Mini Browser to interact with the captive portal.
- Restricting Vulnerable Outbound Traffic (NETBIOS, LDAP).
- Flexible Outbound Connection or Service Allow List.
- DNS Cache Flush or Cache Disabling before VPN connection establishes.
- The ability to disable IPv6
- Limiting the time that a host must pass the captive portal.
 - The VPN must disable all communication with the network once the timeout is reached and no Internet access is detected.
- Preventing the user from accepting invalid VPN server certificates.
- Preventing the end user from disabling the VPN agent.

Here is a mapping of these features per VPN we tested:

( = feature present)

Feature	VPN1	VPN2	VPN3	VPN4
Captive Portal Mini Browser			-	-
Restrict Vulnerable Outbound Traffic	-			
Outbound Connection Allow List	-			
DNS Cache Disable or Flush				
IPv6 Disable				-
Captive Portal Mitigation timeout				-
Prevent user accepting bad VPN server certificate	-			
Prevent user from disabling VPN agent				

VPN1

VPN1 has an explicit mode that is described as ‘lockdown’. Lockdown Mode provides a solution to the captive portal detection and browser interaction phase. The VPN1 client will start an embedded window that is not the default web browser. Once the captive portal authentication process completes, VPN1 initiates the VPN session.

The VPN1 client blocks all applications from initiating network connectivity until the VPN session has established. However, VPN1 has an exception list that allows local network connectivity if processes communicate on ports in the exclusion list.

On Windows, network traffic is locally permitted for the following exception list:

- UDP/TCP port 88 (Kerberos)
- UDP/TCP port 389 (LDAP)
- TCP port 636 (LDAPS)
- TCP port 445 (NETBIOS)
- UDP port 67,68,547,546, (DHCP)
- TCP port 135 (RPC)
- TCP port 3268 (Global Catalog)
- UDP/TCP port 53 (DNS)
- UDP port 5353 (Multicast DNS)

A similar exception list exists for macOS and includes the following ports:

- UDP/TCP port 53 and 5353 (mDNSResponder)
- UDP/TCP port 123 (sntp)
- UDP/TCP port 137-139 (NetAuthAgent)
- UDP/TCP port 111 (kernel_task)
- UDP/TCP port 88 (kcm)
- UDP/TCP port 389, 464, 636, 3268, 3269 (opendirectoryd)

Looking just at the Windows exception list we can see that even though the guest host cannot initiate its HTTP or HTTPS connections, it can communicate on the local network using SMB. This allows the attacker to target NTLM hashes with Responder.

IPv6 data broadcasts were observed in captured and online states for standard mode configuration state. With configuration set to lockdown mode no IPv6 packets were leaked.

VPN2

The VPN2 client presents users with an embedded hotspot browser when it detects the captive portal. The lockdown mode available can be initiated manually but is intended to kick in automatically when it detects a captive portal.

The VPN2 client prevents all network communication except:

- TCP ports 80, 443, and 8080
- UDP port 67,68,547,546, (DHCP)
- UDP/TCP port 53 (DNS)

- UDP port 5353 (Multicast DNS)

The first set of TCP ports listed is configurable. It can be limited to only one port or expanded to include any port.

Once the user completes registration with the captive portal the VPN session is established. During this time, all traffic is directed over the VPN including DNS requests. However, there is still opportunity for DNS leakage if the DNS service associated with the VPN times out or is unable to resolve a host name.

VPN2 does not limit IPv6 communication out of the box. It is still possible to communicate with the guest using IPv6 while all the IPv4 traffic is routed over the VPN. This was observed as part of the Standard Mode configuration for both captured and online connection states. It is possible to disable IPv6 explicitly as part of the lockdown mode configuration to pass this test for both captured and standard mode configurations.

VPN3

VPN3 has several fine-grained configuration options for managing the interaction with captive portals. This results in a very rich and possibly complex set of configuration options. The defaults are sufficient with minimal tweaking necessary to get an acceptable lockdown configuration.

Out of the box the VPN settings default to full tunnelling with Always-On VPN. DNS cache flushing is enforced once the VPN establishes and this feature cannot be disabled.

VPN3 has an option that allows interaction with the LAN of the captive portal even if full VPN tunnelling is used. This feature needs to be disabled as it is enabled by default.

VPN3 does not have a Captive Portal Mini Browser that can be used to interact with the captive portal when in the captured state. The default web browser must be used for this.

IPv6 must be explicitly disabled.

VPN4

This VPN functionality is split across two distinct configuration sets that are managed by two separate applications. It is possible to have a single management configuration interface, but this will require an additional component to be licensed.

VPN4 supports full tunnelling out of the box. VPN4 offers DNS cache manipulation options that include flushing of the cache or even disabling of the Windows DNS cache while the VPN is established.

VPN4 does not have a Captive Portal Mini Browser feature and relies on the default web browser of the host.

IPv6 can only be disabled if the IPSec VPN configuration is used.

Summary of Findings

In summary, our experiments demonstrate that our initial concerns about the failure of VPNs to protect machines in captive portals all hold true. This is not to say that these VPNs do not ‘work’, or that they have ‘bugs’, but rather that captive portals present a use case that VPNs were simply not originally designed to deal with.

Under the assumption that any ‘free’ Wi-Fi service should reasonably be considered malicious, and with an appreciation of contemporary attack vectors and tools, this inability to deal with a significant new use case represents a serious limitation. It forces us to depend on secondary mechanisms like SSL/TLS, firewalls, and endpoint protection to defend the remote endpoint.

The specifics of these features vary from product to product, but generally come down to

- protecting the browser that connects to the portal, and
- limiting the amount of traffic that can leave the computer.

We therefore proceeded to test the VPN products that offered these features with the full capabilities enabled, to determine how effective their protection was.

We were further disappointed to discover that even once fully established, a carelessly configured VPN barely does better at mitigating these very real threats.

In response to the challenges introduced by captive portals, enterprise VPNs have introduced a set of ‘lockdown’ features that are intended to ‘mitigate’ the captive portal problems. These features do indeed address some issues, but unfortunately barely put a dent in the full set of threats we considered for our experiments.

While the behaviour of some of these features have at times perplexed us, we must emphasize that this is once again a fundamental function of how captive portals work, rather than a problem with the products themselves.

The threats we considered in our experiments are by no means catastrophic in nature. Several factors must coincide for the weaknesses to be exploited, and several external factors could prevent such attacks from succeeding.

However, we assert that there is a realistic set of conditions under which modern VPNs fundamentally cannot fulfil their declared objective of securing confidentiality, integrity, and dependable access control.

As our own first-hand experience illustrates, the conditions required to maliciously exploit this weakness in VPN technologies can occur under common real-world circumstances and is probably much more common than we realize.

We would assert that the threat is serious and realistic enough to warrant a serious response by enterprise IT teams, as we will discuss below.

Recommendations

We believe that the vulnerabilities and threats described in these experiments are serious enough to warrant an urgent response, though this need not be expensive or disruptive.

Our technical recommendations can be summarized as follows:

Configuration changes:

- Ensure that you understand and apply the extended configuration options provided by the VPN product with a specific view on the threats highlighted in this paper.
- Avoid using split tunnelling in your VPN configuration. Rather have corporate users' tunnel through the enterprise network where they can be subject to egress filtering, monitoring and other protections the internal network offers.
- Use your VPN configuration to enforce an internal DNS server under your control, and to hardcode the DNS Domain Search Suffix. Both the enterprise VPN products we tested offered this feature, and we expect other serious products to do so also.
- Understand and implement whatever 'lockdown' and 'captive portal mitigation' features your VPN offers. This will not be a simple change and will require careful testing and deployment.
- IPv6 is often overlooked as part of the threat model. It requires a complete review of existing models to understand the impact on mobile devices. The simplest approach could be to disable IPv6 until a verified solution is in place.

Other technical controls:

- Use fully qualified host names everywhere. For example, consistently use 'ocd-src-server.ocd.local' and not just 'ocd-src-server'.
- Local host firewalls and sophisticated Endpoint Detection & Protection programs, properly used, can offer significant defence against the attacks described here.

Strategic thinking:

'If you're not the customer you're the product' is a saying that is frequently used these days.

We believe it holds true for so-called 'free' Wi-Fi services also. The cost to privacy and security that must be offered in exchange for free Internet for mobile users, is to our thinking too high for modern businesses who must take both essentials seriously. We therefore recommend that businesses equip mobile workers with appropriate mobile data technologies and bandwidth so that they can connect via a relatively trustworthy, visible and accountable mobile network provider, rather than a veritable smorgasbord of wholly unknown free Internet providers, whose integrity and motives can never be fully trusted.

Consider Zero Trust:

Zero Trust is an emerging security paradigm in which all networks are considered equal, and untrusted, where there is no internal or external space, and where security must therefore be achieved on the endpoint and on the server without requiring a VPN.

Zero Trust is a security ideology conceived for the modern Internet and being adopted by leading thinkers like Google in their own security strategy. We recommend our customers seriously engage with the Zero Trust concept and the new set of technologies and approaches it advocates, if security is to remain relevant in the face of changing technologies and emerging threats over the next five to ten years.

Conclusion

Security technologies emerge onto the market in response to a specific set of threats.

As the needs of the client and the technology landscape evolve, however, so must the security product. Ensuring continued alignment between evolving threats and the technologies we use to mitigate them requires constant vigilance.

Our investigation regarding the effectiveness of VPN products in the context of modern Internet configurations raises significant cause for concern. The issue is really a larger one, however, regarding the constant effort required to understand the threat, the difficulty of understanding how our security tools align to the threat, and ultimately ensuring that we are using those tools to their full effect. No technology on its own makes a problem go away.

Relying on VPNs alone is clearly not enough. Trade-offs such as usability and remote support seems to outweigh hardened security. Sensibly limiting host interaction with network infrastructure that does not lead to exploitation is challenging and requires a lot of testing and verification.

That is our responsibility, and it has not gotten any easier.

Glossary

DNS Search Suffix	The domain name that the host uses when resolving unqualified host names. This value is attached to the end of the value to form a fully qualified domain name. This value can also be specified as part of a list. RFC 2132 option 15 and RFC 3397 option 119 contains the technical description of this in terms of DHCP parameters.
Pass-the-hash	A hacking technique that involves stealing an NTLM or LanMan cryptographic hash that is representation of a user's password instead of the clear text equivalent. This hash value is used to authenticate as another user, thus impersonating them.
Person-in-the-middle	Also known as man-in-the-middle. A type of attack where the attacker inserts themselves between two parties without either knowing that the attacker is intercepting, recording, and possibly modifying the data exchanged between the given parties.
PiTM	See Person-in-the-middle.
Captive Portal	A special kind of Wi-Fi access point that offers Internet access. Internet access is granted once the guest provided consent, agreed to terms and conditions, successfully provided proof of payment, or completed an authentication process. The interaction is normally done using a web server, thus requiring the client to have a web browser to interact with the web site.
Captive Portal Mini Browser	A CPMB is a sandbox mini web browser that handles the interaction with the captive portal web site. It is discarded once Internet access is obtain or if a timeout period is reached.
Captive Portal Remediation	Controls that are put in place to ensure that the host connecting to a Wi-Fi network, with or with-out a captive portal, is protected from attacks while joined to this Wi-Fi network.
Captive Portal Post-Remediation	After Captive Portal Remediation is applied.
Captive Portal Pre-Remediation	Before Captive Portal Remediation is applied.

DNS Proxying	A Domain Name Server (DNS) that fronts other DNS servers with the possibility to modify response values.
Hotspot	An alternate name for captive portal.
Lockdown Mode	VPN configuration that lessens the exposure to threats while connected to a Captive Portal in either the Capture or Online state.
Responder	A tool originally created by Trustwave's SpiderLabs. Responder is a LLMNR, NBT-NS and MDNS poisoner, with built-in HTTP/SMB/MSSQL/FTP/LDAP rogue authentication server supporting NTLMv1/NTLMv2/LMv2, Extended Security NTLMSSP and Basic HTTP authentication.
Split Tunnel	When a VPN is configured to route specific network requests through the VPN tunnel while other traffic follows default network routing rules. The latter may result in Internet access.
VPN	Virtual Private Network. In our context also referred to a Secure Remote Access.