Andrew G. Watters, Esq.
Litigation and Trial Law Firm
+1 (650) 542-0057

555 Twin Dolphin Dr., Ste. 135
Redwood City, CA 94065
andrew@watters.law

SMCBA I.P./TECHNOLOGY SECTION
Workshop: Create Your Own Website From Scratch
January 7, 2025

I. Introduction

Today's program provides a brief history of the technologies that underlay the World Wide Web, followed by a crash course in the present-day Linux command line and ViM, concluding with the creation of attendees' own web pages using the skills learned today.

Terminal-Based Operating Systems - 1960-present

Until the 1960s, computers required punch cards and hard-copy teletype-type printers to interface with the user. In the 1960s, computer terminals with video displays were invented. In a common configuration, one mainframe computer would have several computer terminals connected to it. Users of the terminals would share computing resources on the mainframe, such as processor and memory. This client-server model is directly analogous to the present day, in which users interact with servers primarily through "terminal emulators" that mimic the behavior of physical terminals. The following figure shows a Digital Equipment Corp. VT100 terminal, circa 1978, connected to a DEC PDP-11 mainframe and displaying a simple list of files. The VT100 was a bestseller in its era. A key takeaway from the terminal era is that the computing was done on the mainframe/server, with the terminal simply acting as a display and controller.

Coinciding with the era of video terminals, a common problem was that more people wanted to share time on the terminals than physical terminals were available; however, the mainframe was still idle for a substantial percentage of the time even with all of the physical terminals occupied. This prompted the creation of Multics, Unix, and Telnet in the 1969 time frame. Multics and Unix were multi-user operating systems intended to share resources on one machine across multiple users. Telnet allowed someone with a user account on a mainframe to log into the mainframe from other than a physical terminal, such as another mainframe or computer in some other location connected to a physical network. This concept of sharing time on a server from locations without direct physical access to the computer is the concept and need behind ARPAnet. ARPAnet was the precursor of today's Internet. ARPA was the Advanced Research Projects Agency of the U.S. Government (now called DARPA). A grant from ARPA allowed several computer scientists at UCLA to create the first operational wide-area packet-switched computer network, which went online in 1971. Packet switching means that individual packets forming the communication are split up by the sender's computer, directed to the recipient as resources permit, and are reassembled by the recipient's computer in the correct order to present a complete communication. This is unlike circuit switching as used in the 1970s telephone network, which required a full allocation of bandwidth for the duration of the transmission and was less efficient. With packet switching, a transmission line can be used at 100% of its capacity and still complete all communications intact, therefore you need less bandwidth reserved for each individual sender and recipient. Think of it like cars on a freeway and you'll see that if all the cars travel at the same speed, the freeway can reach capacity and still allow all the cars to reach their destinations. This is unlike an airplane with a limited number of seats that they overbook, which is essentially the cause of your telephone busy signal when all circuits are occupied.

The Internet Protocol Suite - 1974

Packet switching was formalized in 1974 with the Transmission Control Protocol/Internet Protocol Suite (TCP/IP), which quickly became the predominant standard. TCP/IP provides the ability for packets to be transmitted between remote locations by tagging them with sequence and flag information, headers, and contents. The headers would show, for example, the destination I.P. address of the recipient. Internet Protocol version 4 (IPv4) is still used today, and consists of a sequence of four numbers that uniquely identify the network on which the computer is located. For example, the main IPv4 address of my office computer system is 38.122.223.130. This can be reversed and is still a valid address for some purposes, such as when creating DNS zones: 130.223.122.38.in-addr.arpa. There are whole multi-volume treatises on TCP/IP and its use over the last 50 years. All you need to know for today is that IP addresses (both IPv4 and IPv6) are the network addresses of computer systems on the Internet (and local-area networks (LANs) when using a private IP address range). The IP address range allowed by the Internet Assigned Numbers Authority (IANA) generally is 1-255.1-255.1-255.1-255, which means that the world has an IPv4 address space of 4,294,967,296 addresses. This address space has long-since been exhausted, which has presented a number of solutions, such as Network Address Translation (NAT) and IPv6. IPv6 is vastly superior, but is more difficult to

understand and configure, which has been hindering its adoption. The IPv6 address of my office hardware is 2001:550:2:b::56:2, which shows the different IPv6 format. There is an essentially unlimited number of IPv6 addresses due to the design of the system, which anticipates a larger world with universal connectivity, unlike IPv4. In any case, the adoption of TCP/IP packet switching was sealed when, in 1982, the U.S. government adopted it as the standard for Department of Defense networking.

Modern Email (SMTP) - 1983

Email had sort-of existed from 1971, when a researcher sent the first email using the @ symbol. The Simple Mail Transfer Protocol was adopted in 1983 and provided a standard, open standard for sending and receiving emails that still exists today in modernized form. Email is important because it caused the creation of the Domain Name System (DNS). Before DNS, emails had to be sent to the IP address of the server where the user had the email account. Obviously, it's hard to remember a sequence of four numbers when sending an email, but a lot easier to remember something like "cs@berkeley.edu."

Modern Domain Name System (BIND) - 1984

Speaking of Berkeley, the Berkeley Internet Name Domain (BIND) system was developed there in the early 1980s, also with a grant from DARPA. In the BIND model, the Internet is divided into DNS "zones" that each have responsibility for a certain name space. For example, the Berkeley zone requires UC Berkeley to have a DNS server that answers queries authoritatively for the Berkeley range of IP addresses. This is seen in the following command on the macOS terminal using the "dig" utility (Domain Information Groper[1]) to query the Start of Authority (SOA) for berkeley.edu:

```
raellic@Andrews-MacBook-Pro ~ % dig soa berkeley.edu

; <<>> DiG 9.10.6 <<>> soa berkeley.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21904
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1432
;; QUESTION SECTION:
;berkeley.edu. IN SOA

;; ANSWER SECTION:
berkeley.edu. 10800 IN SOA sut-vps-ib1.net.berkeley.edu. hostmaster.berkeley.edu.
2063461601 10800 1080 2419200 300
```

[1]As with many computer acronyms, dig was coined before the #MeToo era, and its name reflects the in-joke and lack of sensitivity that is common for late-20th century computer engineering.

```
;; Query time: 20 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Tue Dec 31 12:06:21 PST 2024
;; MSG SIZE  rcvd: 104

raellic@Andrews-MacBook-Pro ~ %
```

In this query, sut-vps-ib1.net.berkeley.edu is the authoritative nameserver for the berkeley.edu zone. If we query that nameserver, we can determine, for example, where the email server for berkeley.edu is by asking the Berkeley DNS server where email should be sent (MX record):

```
raellic@Andrews-MacBook-Pro ~ % dig @sut-vps-ib1.net.berkeley.edu mx berkeley.edu

; <<>> DiG 9.10.6 <<>> @sut-vps-ib1.net.berkeley.edu mx berkeley.edu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5413
;; flags: qr aa rd; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1220
;; QUESTION SECTION:
;berkeley.edu. IN MX

;; ANSWER SECTION:
berkeley.edu. 271 IN MX 10 alt3.aspmx.l.google.com.
berkeley.edu. 271 IN MX 10 alt4.aspmx.l.google.com.
berkeley.edu. 271 IN MX 1 aspmx.l.google.com.
berkeley.edu. 271 IN MX 5 alt2.aspmx.l.google.com.
berkeley.edu. 271 IN MX 5 alt1.aspmx.l.google.com.

;; Query time: 20 msec
;; SERVER: 136.152.1.39#53(136.152.1.39)
;; WHEN: Tue Dec 31 12:11:34 PST 2024
;; MSG SIZE  rcvd: 159

raellic@Andrews-MacBook-Pro ~ %
```

From this, we have determined that UC Berkeley uses Gmail for its email for all berkeley.edu addresses. Note that other subdomains may have internal email systems, or some other configuration that is not visible to the outside internet without investigating further. We can also determine where the Berkeley DNS server resides by inquiring as to

its IP address and then executing a whois command:

```
raellic@Andrews-MacBook-Pro ~ % dig sut-vps-ib1.net.berkeley.edu

; <<>> DiG 9.10.6 <<>> sut-vps-ib1.net.berkeley.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29348
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1432
;; QUESTION SECTION:
;sut-vps-ib1.net.berkeley.edu. IN A

;; ANSWER SECTION:
sut-vps-ib1.net.berkeley.edu. 10619 IN A 136.152.1.39

;; Query time: 11 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Tue Dec 31 12:14:35 PST 2024
;; MSG SIZE  rcvd: 73

raellic@Andrews-MacBook-Pro ~ % whois 136.152.1.39
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

refer:        whois.arin.net

inetnum:      136.0.0.0 - 136.255.255.255
organisation: Administered by ARIN
status:       LEGACY

whois:        whois.arin.net

changed:      1993-05
source:       IANA

# whois.arin.net

NetRange:       136.152.0.0 - 136.152.255.255
CIDR:           136.152.0.0/16
NetName:        UCB-TELECOM
NetHandle:      NET-136-152-0-0-1
```

```
Parent:          NET136 (NET-136-0-0-0-0)
NetType:         Direct Allocation
OriginAS:        AS25
Organization:    University of California at Berkeley (UCAB-1-Z)
RegDate:         1991-03-06
Updated:         2023-12-14
Comment:         DMCA Designated Agent is dmca@berkeley.edu
Ref:             https://rdap.arin.net/registry/ip/136.152.0.0


OrgName:         University of California at Berkeley
OrgId:           UCAB-1-Z
Address:         IST - Telecommunications
Address:         ATTN Network Services and Operations
Address:         2850 Telegraph Ave
Address:         NOTE See Comment for DMCA INFO
City:            Berkeley
StateProv:       CA
PostalCode:      94705
Country:         US
RegDate:         2010-01-13
Updated:         2023-07-20
Comment:         DMCA Designated Agent is dmca@berkeley.edu
Ref:             https://rdap.arin.net/registry/entity/UCAB-1-Z


OrgNOCHandle: UCB-NOC-ARIN
OrgNOCName:   IST Communication and Network Services
OrgNOCPhone:  +1-510-664-9000
OrgNOCEmail:  noc@berkeley.edu
OrgNOCRef:    https://rdap.arin.net/registry/entity/UCB-NOC-ARIN

OrgTechHandle: UCB-NOC-ARIN
OrgTechName:   IST Communication and Network Services
OrgTechPhone:  +1-510-664-9000
OrgTechEmail:  noc@berkeley.edu
OrgTechRef:    https://rdap.arin.net/registry/entity/UCB-NOC-ARIN

OrgAbuseHandle: UCBSE-ARIN
OrgAbuseName:   UCB-SECURIT-ARIN
OrgAbusePhone:  +1-510-664-9000
OrgAbuseEmail:  abuse@security.berkeley.edu
OrgAbuseRef:    https://rdap.arin.net/registry/entity/UCBSE-ARIN

RTechHandle: UCB-NOC-ARIN
```

```
RTechName:   IST Communication and Network Services
RTechPhone:  +1-510-664-9000
RTechEmail:  noc@berkeley.edu
RTechRef:    https://rdap.arin.net/registry/entity/UCB-NOC-ARIN

raellic@Andrews-MacBook-Pro ~ %
```

From this, we can determine that the Berkeley DNS server likely resides on campus at UC Berkeley, because it is within the IP address range for UC Berkeley (136.0.0.0 - 136.255.255.255). The point is that every domain name must have an authoritative nameserver that provides other computers on the Internet a link between the text-based domain names and the numerical IP addresses within that zone. Let's try a dig query on my own equipment:

```
raellic@Andrews-MacBook-Pro ~ % dig www.andrewwatters.com

; <<>> DiG 9.10.6 <<>> www.andrewwatters.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38539
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1432
;; QUESTION SECTION:
;www.andrewwatters.com. IN A

;; ANSWER SECTION:
www.andrewwatters.com. 939 IN A 38.122.223.130

;; Query time: 10 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Tue Dec 31 12:17:37 PST 2024
;; MSG SIZE  rcvd: 66

raellic@Andrews-MacBook-Pro ~ %
```

This shows that my website is hosted at the IP address indicated in the dig output above. More on that later. In summary, DNS allows users to match up domain names with IP addresses, which is critical for sending emails so people don't have to remember IP addresses, which also may change over time. DNS greatly simplifies a range of services that form the Internet, such as now-disused protocols like Gopher, Fido, FTP, Telnet, and others that have been crushed by the World Wide Web.

The World Wide Web - 1989

In the late 1980s, Lotus, IBM, Novell, and a host of other companies had created proprietary systems that allowed users in a corporate environment to collaborate with each other over a local area network (LAN), such as Lotus Notes, Novell NetWare, and so on. For consumer use, there was also America Online (AOL), CompuServe, Prodigy, and other Internet Service Providers (ISPs). But there was no World Wide Web (WWW) yet. Each of the aforementioned systems was proprietary and required users to log into their ISP's own portal. The World Wide Web was the first publicly-accessible Internet portal, so to speak. The World Wide Web was invented in 1989 at CERN, which is the European Agency for Nuclear Research. Tim Berners-Lee, a scientist at CERN, created the world's first webserver on his NeXT workstation by programming it to deliver pages over the CERN local area network (LAN) to other users. He also created the first version of Hypertext Markup Language (HTML) to "link" web pages to other pages, and embed resources such as images into web pages. The world's first web page looked like this when viewed from a terminal:

The first proper website from around 1991 is still online (on other equipment) at CERN: http://info.cern.ch/hypertext/WWW/TheProject.html

The point of all this is that the convergence of the above history and technologies enabled and created the World Wide Web, which over just a handful of years emerged as the world's preferred method of consuming information. TCP/IP, DNS, and the then-new Hypertext Transfer Protocol (HTTP) are still with us today, in updated form.

Creation of Linux - 1991

Linus Torvalds, then a computer science student at the University of Helsinki, created

Linux in 1991 as an alternative to Minix, which was itself a free version of Unix. Over the last 33 years, Linux has crushed all other rivals in the server space, and is the predominant operating system used by web servers across the world today. Linux replicates many of the features of Unix in a free or low-cost setup, and there are many variants of Linux from numerous vendors. I personally use Red Hat Enterprise Linux (RHEL), which is marketed as a stable, conservative version of Linux suitable for enterprise use. Linux enabled the mass creation of web servers in the '90s because otherwise, you were spending $25,000 or more for a Unix license for a system outside the university context.

SSH and SSL/TLS - 1995

Until 1995, when Secure Shell (SSH) was invented, it was not possible to securely log into another computer across public infrastructure such as the internet. Anyone with access to the channel could read usernames and passwords in plaintext. SSH, with its free OpenSSH library, enabled the mass remote management of command line systems, including Linux, securely and for free. 1995 also saw the Secure Sockets Layer (SSL) certificate concept standardized, which allowed websites to encrypt information between the server and the end user over public transmission systems. This allowed online shopping with Amazon, which also changed the world. TLS, which stands for Transport Layer Security, also emerged in this era.

Mass Adoption Through Today - 1995 to present

I like to think of the World Wide Web as viewed in distinct eras depending on who was President at the time. This doesn't mean they had anything to do with it, it just means that those are convenient blocks of time. It should put things in perspective to say that when the World Wide Web was invented, George H.W. Bush was President. Linux was invented before Bill Clinton took office. Bulletin Board Systems (BBSes) such as AOL and CompuServe were common in the same era. I received a 14.4k modem as a birthday gift in 1993, which I used to log into BBSes. I personally was introduced to the World Wide Web by my brother, now a software engineer, in approximately 1995. The Clinton Presidency saw the mass adoption of the World Wide Web for all kinds of uses through the "dot com" boom of 1998-2000. The George W. Bush presidency is when the World Wide Web truly exploded, in my opinion. Using the "web" became normal for regular people outside of the education context. And by the time Barack Obama took office in 2008, the web was the world's largest and easiest system from which to consume information. The information superhighway that Al Gore and others championed in Congress finally came to fruition during the Obama era. Since then, the mass market availability of web hosting and the costs of hosting approaching zero, plus the ease of using pre-programmed platforms such as Wordpress, have left today's web in a state of perhaps ease of creation and ease of consumption, but without a fundamental understanding of how these systems actually work. Something was lost along the way, in my opinion, which is why I make it a point to hand-code my websites and web applications from the Linux command line. I understand how this works, which presents a lot of benefits, like how to troubleshoot emergencies and how to fix broken websites.

II. Linux Crash Course

Linux as used in web servers is a text-based multi-user operating system that relies on a terminal emulator to accomplish work. You can also set up a graphical user interface if you desire a Linux workstation, which I have in my office as my daily driver. Today's program focuses on practical use of the skills needed to create a website on a Linux server. To that end, I've provided you with login credentials for my live web server, and access to a publicly viewable directory on the server. We are going to do a practical exercise in which you create your own folder to contain your website. The following steps will be demonstrated:

1. Open the terminal program.
2. Log into my web server with SSH.
3. List the contents of the home directory (folder).
4. Note the mode/permissions of each file and directory.
5. Note the user and group of each file and directory.
6. Change directories to the web directory.
7. Create your own folder.
8. Enter your own folder.
9. Read manual pages for different commands that might be useful: ls, mkdir, ln, touch, head, tail, and history.
10. Put it all together before creating a web page: create an empty file in the correct directory with the correct permissions.

III. ViM Crash Course

1. View/selection mode.
2. Navigation, including by-word (w), reverse-by-word (b), beginning (0), and end ($).
3. Set number, syntax on.
4. Insert mode (i).
5. Visual mode (v).
6. Cut, copy, and paste.
7. Command mode.
8. Write file and quit.

IV. Create Your Own Website

1. Discussion of name-based virtual hosting.
2. Live demonstration of creating an index page on watters.law.
3. Embed an image in your page.

4. Create a second page to link from the index page.
5. Link the pages to each other.
6. Continue editing with your bio or other information.
7. Zip the files into the same folder.
8. Email yourself the package for later use.
9. Case study: updating my news section on my website.

V. Conclusion

Today's workshop provided a background of the technologies that form the basis for the World Wide Web, a crash course in Linux commands and ViM, and your own websites to take home. Thank you for attending, and please enjoy some well-deserved beverages in my office.