The internet at 40
Pioneers: Weld Pond and Mudge

In 1998, seven young men were invited to testify to the senate government affairs committee about the state of the US government's computer network. They were members of The Lopht, a group of internet pioneers who searched for weaknesses in computer systems and computer software – in other words, they were hackers. At around the same time, two members of The Lopht, Mudge and Weld Pond, were interviewed for an Open University programme called Cyberwars.

The Lopht is seven unique individuals who have been called hackers. We've been called troublemakers. We've been called solution providers. It's just a bunch of kids who are interested in figuring out how things work in technology. So we rip them apart and find the flaws and try and share and the information we find with everybody else.

Originally the phrase hacker really came out of the old Massachusetts Institute of Technology and it was just simply to do something unique and ingenious in a novel way. Any clever way of doing something was called a hack. Unfortunately a few years down the road the media labelled the phrase hacker as a very derogatory term to somebody who breaks into systems and causes trouble, but actual vandalism or trouble making really wasn't part of it. That was a media adopted phrase. A lot of times in the hacker community we refer to things like that such as crackers or you know maybe pranksters or something.

The people who use these computer systems need to know the vulnerability's there if they wait for the manufacturer to tell them they're going to be vulnerable a lot longer.

The main reason we would target a particular operating system like MicrosoftNT is we see that it's being adopted by lots of people. We see that it's being sold as a secure solution and whenever someone is selling a secure solution and that's what's coming out of a marketing department, you know our ears perk up and we say well is it really secure? And so we just start banging on something that's popular and being sold as secure we just bang on it and as soon as we start finding flaws and problems with it that just eggs us on to keep going. And it's interesting to watch the manufacturer of the software how they react to us.

Whenever we talk about a flaw or vulnerability we are not going after any individual user. We are not saying oh look at City Bank they have this web server set up and if you just type in this password you get right in. We're not or you know because there's a back door or something. We don't talk about things like that but we might say 'Microsoft's web server has this flaw and all the users who use this web server need to know about it'. So I think there's a big difference there.

I mean this is just classic in the software industry. You put it out there and then when problems are found by your users then you fix the problems. If no one finds it I guess it wasn't a big problem. But with security issues – I mean you now that's a whole class of problems that are solvable but with security things it's especially bad because as soon as that problem is found you know, things of real value, we're not talking about you know I said make it red and bold and it came out purple – we're talking about things where people's whole businesses are running on this software and if a problem's found and something which can totally shut someone's business down you know that's a lot of – that's you know rather serious.

Think about buying a bullet proof vest. Would you like the company to go and say here's a bullet proof vest. We really just want to sell a bunch of them so it might not be tested tremendously well. If somebody finds a vulnerability in it later maybe we'll go back and fix it.

But just you know wear it and feel comfortable right now. No way! I mean that's what they're doing with the software out on the networks. They're saying here's our secure web server; here's our secure electronic commerce transactions, we really would much rather write the cheapest most cost effective code we don't need to have people who are security experts because that costs more money than just having regular people who just churn out code left and right. Well, legally in the United States right now at least the software companies can just say oops sorry – and that's all that they're liable for.

Usually when you install a software programme there's a screen that comes up which is a licence agreement which is usually like five or six pages of legalise and no one really reads that or understands it. They just say okay because they want to run the programme. Well in that agreement it says if this software fails for any reason usually your only recourse is to get a refund.

Mudge, Weld and the other Lopht members worked in a scientific way and made sure they always stayed within the law.

We have probably about fifty different machines set up in here on networks. If we want to break in to a system we'll set it up locally here and attack it. There are a couple of advantages to this. One it keep us out of jail because we are not breaking into somebody else's systems that we don't own and don't legitimately have access to. And the other thing is it's in a controlled environment and just like any research environment you want to be able to control everything else that's happening around it. Lets say that you wanted to look at a problem in one of Microsoft's web servers or one of Lotus' databases. You could go and use one that's publicly available on the network but you don't know who else is using it or what other interactions might be affecting your tests. If you set it up internally you control the entire environment so you know when you plug data in and you get data out that was a direct result of your experiment so it makes it much quicker and much easier to find the actual flaws. Once you find them in this environment they'll work in any other environment. But you don't have to worry about the noise generated by other interactions.

Like all the members of the Lopht, Mudge and Weld didn't use their real names

We find that often we have to work with pseudonyms because of the type of work we do and the way that it offends certain companies. If you're a multimillion dollar company or a multibillion dollar company and seven individuals here cost you a couple of points on the stock market, not out of any sort of malice and not out of trying to profiteer on it, but basically by showing that you are selling snake oil, or that your product does not behave or operate the way they claimed it did. We've had situations where they've expressed interest in making our lives a little more difficult. With the pseudonyms it's another layer of abstraction. It's not preventing people from actually finding out who we are. But it stops them from going to the companies that we work for right off the bat; especially if the companies we work for would have relationships with the organisation we offended. We know several organisations where people who don't operate under pseudonyms - and this is all done in our spare time, this isn't done for any organisation or company - but we know some people who in their spare time did some research that offended a couple of large companies and that companies legal team approached the company that these people worked for and said if you know either fire him or we're gonna sue. And we'll stop you because we're a much bigger company. We don't need that that difficulty. If they want to come after us they - they come after us here because we have no money so it's not worth it for them to sue us and that would turn us into huge martyr heroes and that would just shove the problem that we had exposed out you know to that many more people. So they kind of leave us alone. It's a little Robin Hood.